



# A Survey on Deep-Learning-Based Real-Time SAR Ship Detection

Jianwei Li , Jie Chen , Pu Cheng, Zhentao Yu, Lu Yu, and Cheng Chi

**Abstract**—Recently, deep learning has greatly promoted the development of synthetic aperture radar (SAR) ship detection. But the detectors are usually heavy and computation intensive, which hinder the usage on the edge. In order to solve this problem, a lot of lightweight networks and acceleration ideas are proposed. In this survey, we review the papers that are about real-time SAR ship detection. We first introduce the model compression and acceleration methods. They are pruning, quantization, knowledge distillation, low-rank factorization, lightweight networks, and model deployment. They are the source of innovation in real-time SAR ship detection. Then, we summarize the real-time object detection methods. They are two-stage, single-stage, anchor free, trained from scratch, model compression, and acceleration. Researchers in SAR ship detection usually learn from these ideas. We then spend a lot of content on the review of the 70 real-time SAR ship detection papers. The years, datasets, journals, deep-learning frameworks, and hardwares are introduced first. After that, 10 public datasets and the evaluation metrics are shown. Then, we survey the 70 papers according to anchor free, trained from scratch, YOLO series, constant false alarm rate+convolutional neural network, lightweight backbone, pruning, quantization, knowledge distillation, and hardware deployment. The experimental results show that the algorithms have been greatly developed in speed and accuracy. In the end, we pointed out the problems of 70 papers and the directions to be studied in the future. This article can enable researchers to quickly understand the research status in this field.

**Index Terms**—Anchor free, computer vision, dataset, deep learning, lightweight, real-time, synthetic aperture radar (SAR) ship detection, single-stage detector, trained from scratch.

## I. INTRODUCTION

**S**YNTHETIC aperture radar (SAR) is an all-day and all-weather sensor that is widely used in military, agricultural, geological, ecological, marine, and other fields. SAR ship detection is an important tool for marine monitoring in illegal fishing, oil spill detection, and maritime traffic management [1], [2], [3].

The traditional detection methods are based on constant false alarm rate (CFAR). It firstly models the clutter, and then obtains the threshold value according to the false alarm rate. The pixels above the threshold are regarded as ship pixels, and those below the threshold are regarded as background. The performance of this method largely depends on the statistical modeling of sea

Manuscript received 3 December 2022; revised 3 February 2023; accepted 7 February 2023. Date of publication 1 March 2023; date of current version 6 April 2023. This work was supported by the National Natural Science Foundation of China under Grant 61790550. (Corresponding author: Jie Chen.)

The authors are with the Naval Submarine Academy, Qingdao 266000, China (e-mail: ljm\_jw@163.com; grass2009@163.com; hagooddemon@163.com; qdqyzt@163.com; yulu\_china@163.com; cheng.chihhu@163.com).

Digital Object Identifier 10.1109/JSTARS.2023.3244616

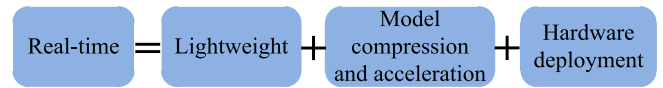


Fig. 1. Lightweight network designing, model compression and acceleration, and hardware deployment are three ways to implement real-time object detection.

clutter and the parameter estimation of the selected model. In general, when the scene is simple, the CFAR method can achieve good result. However, for complex offshore scenarios, it will have more false positives and poor performance [4].

After CFAR, discrimination is used to classify the ship and nonship areas. The features that are used are length, width, HoG, Surf, LBPs, and so on. The classifiers that are used are SVM, MLP, and so on. But since deep learning revives in 2012 [5], the above ideas show disadvantage in accuracy. The deep-learning method works in an end-to-end manner. This means that it does not need to optimize multiple steps individually. It optimizes the whole detection system simultaneously. It can adapt to various complex scenes and has strong robustness [6], [7], [8]. Due to its great advantage, SAR researchers began to use these methods.

The beginning is the emergency of SSDD [9]. Based on it, researchers can train and test their deep-learning-based detectors, and also can compare their algorithms each other in the same evaluation metrics. Since then, a large number of datasets and algorithms have been proposed, further promoting the development of this field. Li et al. [10] reviewed these achievements comprehensively.

The above deep-learning-based detectors are all based on convolutional neural networks (CNN)'s strong feature extraction ability. In order to get good performance, most of the CNNs are deep and wide, which have high computational complexity and memory consumption. For example, AlexNet has 60 million parameters with five convolutional layers and three fully connected layers [11]. It is difficult to deploy such large model on edge devices with limited computing resources and memory. In order to realize real-time SAR ship detection, lightweight network designing, model compression and acceleration, and hardware deployment are three ways researchers usually take, as shown in Fig. 1 [12].

Due to the huge demand of real-time SAR ship detection, a lot of researchers try to realize real-time SAR ship detection with different accuracy and speed tradeoff. As far as we know, there are 70 papers [13], [14], [15], [16], [17], [18], [19], [20], [21], [22], [23], [24], [25], [26], [27], [28], [29], [30], [31], [32], [33],

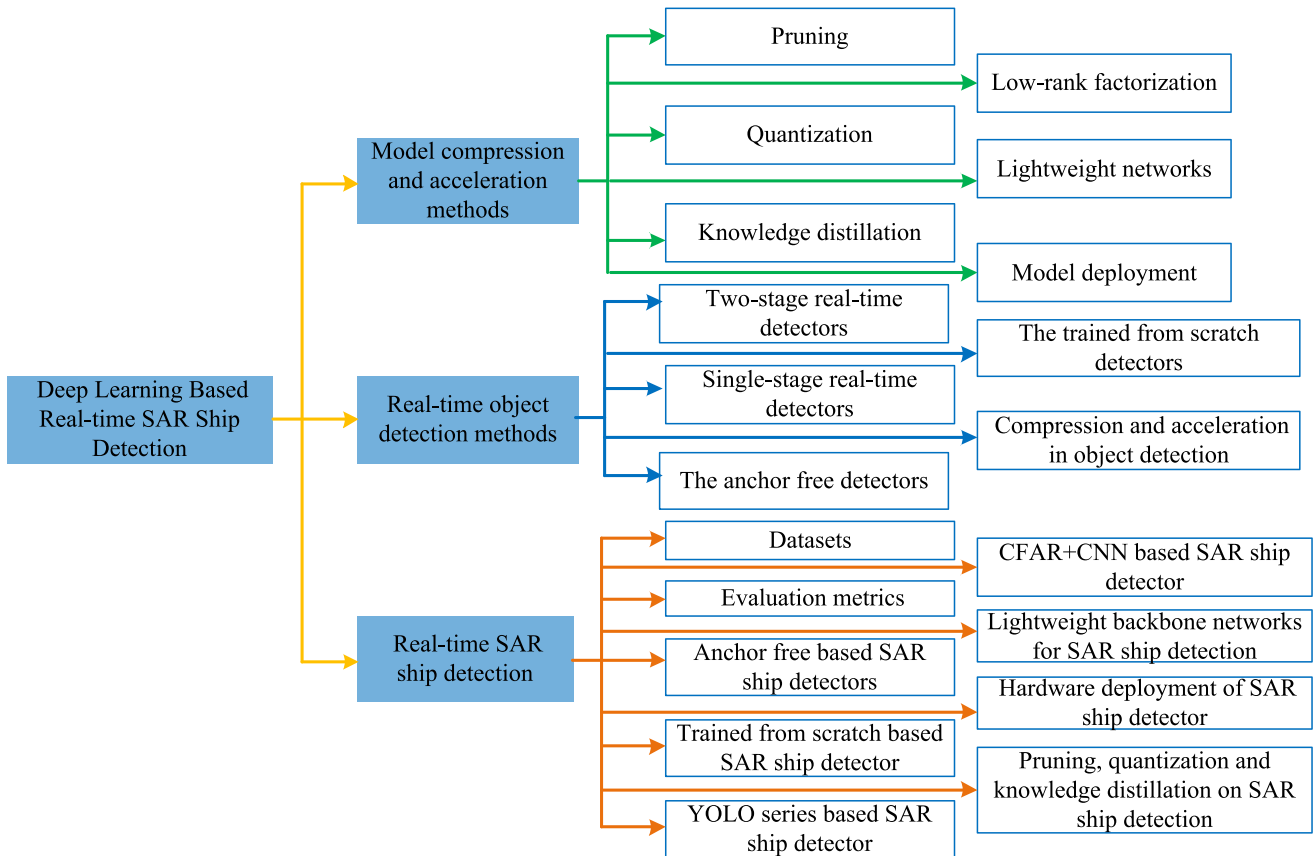


Fig. 2. Overall architecture of the article.

[34], [35], [36], [37], [38], [39], [40], [41], [42], [43], [44], [45], [46], [47], [48], [49], [50], [51], [52], [53], [54], [55], [56], [57], [58], [59], [60], [61], [62], [63], [64], [65], [66], [67], [68], [69], [70], [71], [72], [73], [74], [75], [76], [77], [78], [79], [80], [81], [82] that are about this. Unfortunately, nobody reviews them comprehensively. This is one of the motivations of this article.

The contributions of this article are as follows. Firstly, we summarize the model compression methods, acceleration methods, and the real-time object detection methods. They are often used in real-time SAR ship detection. Secondly, the datasets and the evaluation metrics those are usually are systematically summarized here. Thirdly, 70 papers those are related to real-time SAR ship detection are divided into 7 categories and are reviewed thoroughly. Finally, the future directions are given for the following work. As far as we know, this is the first comprehensive survey in this field.

The rest of this review is arranged as shown in Fig. 2. Section II briefly analyzes some work related to this article. Section III summaries the model compression and acceleration methods. It mainly includes pruning, quantization, knowledge distillation, low-rank factorization, lightweight networks, and model deployment. Section IV introduces the real-time detection methods. It mainly includes two-stage, single-stage, anchor free, trained from scratch, compression, and acceleration. Section V is the main content of the article. We introduce the general situation of the 70 papers. Then the datasets, evaluation metrics, anchor free, trained from scratch, YOLO series, CFAR+CNN, lightweight

backbone, pruning, quantization and knowledge distillation, and hardware deployment are analyzed, respectively. Section VI discusses the problem and the future direction of this field. Section VII is the conclusion of the article.

## II. RELATED WORK

As far as we know, there is no comprehensive paper on real-time SAR ship detection. This is mainly because this field is relatively new, with few achievements, and is difficult to carry out a review. At present, the review papers those are related to this article are shown in Table I.

Mao et al. [83], Stefanowicz et al. [84], Zhang et al. [85], and Li et al. [86] are summaries of SAR ship detection based on deep learning. Mao et al. [83] compared the classical deep-learning-based detectors on SSDD. It provided a benchmark for researchers in this area. In essence, it is not a survey of papers. Jerzy et al. [84] surveyed the papers about SAR ship detection from 2015 to 2020. In this article, the CFAR, CNN, GLRT, feature extraction, weighted information entropy, and variational Bayesian inference methods are studied. The deep-learning-based SAR ship detection is not studied thoroughly compared with other papers. Zhang et al. [85] provided an official version of SSDD. It introduced the drawbacks and success of SSDD. It provided three versions of SSDD. It also provided the seven using standards of the official SSDD. But it only reviewed the papers that are related to SSDD, the papers using other

TABLE I  
PAPERS THOSE ARE RELATED TO OUR WORK

No.	Date	Author	Title	Journal/Conference
83	Jun 20 2020	Yuxing Mao et al.	Ship Detection for SAR Imagery Based on Deep Learning: A Benchmark	2020 IEEE 9th Joint International Information Technology and Artificial Intelligence Conference (ITAIC)
84	Oct 14 2020	Jerzy Stefanowicz et al.	Current trends in ship detection in single polarization synthetic aperture radar imagery	Photonics Applications in Astronomy, Communications, Industry, and High Energy Physics Experiments 2020
85	Sept 15 2021	Tianwen Zhang et al.	SAR Ship Detection Dataset (SSDD): Official Release and Comprehensive Data Analysis	Remote Sensing
86	Jun 5 2022	Jianwei Li et al.	Deep Learning for SAR Ship Detection: Past, Present and Future	Remote Sensing
87	Sept 5 2019	Licheng Jiao et al.	A Survey of Deep Learning-based Object Detection	IEEE ACCESS
88	Sept 26 2019	Li Liu et al.	Deep Learning for Generic Object Detection: A Survey	arXiv
89	Sept 5 2019	Licheng Jiao et al.	A Survey of Deep Learning-Based Object Detection	IEEE ACCESS
90	Oct 23 2017	Yu Cheng et al.	A Survey of Model Compression and Acceleration for Deep Neural Networks	arXiv
91	Oct 13 2020	Abhinav Goel et al.	A Survey of Methods for Low-Power Deep Learning and Computer Vision	2020 IEEE 6th World Forum on Internet of Things (WF-IoT)
92	Oct 5 2020	Rahul Mishra et al.	A Survey on Deep Neural Network Compression: Challenges, Overview, and Solutions	arXiv
93	Mar 3 2021	Ke Zhang et al.	Compacting Deep Neural Networks for Internet of Things: Methods and Applications	IEEE Internet of Things Journal

public datasets are not reviewed. Li et al. [86] finished the first comprehensive survey of SAR ship detection. It analyzed the past, present, and future of this area by the 177 published papers. It can make researchers better understand these algorithms and encourage more researchers to this field. But it only introduced the real-time SAR ship detection roughly, which is not sufficient as real-time SAR ship detection is becoming hot and hot recently.

Jiao et al. [87], Liu et al. [88], and Jiao et al. [89] are summaries of the object detection papers in computer vision, which has a comprehensive summary of the real-time object detection algorithms. The real-time SAR ship detection can learn from these. Cheng et al. [90], Goel et al. [91], Mishra et al. [92], and Zhang et al. [93] are reviews on the compression and acceleration of deep-learning-based computer vision. These achievements point out the direction for realizing real-time SAR ship detection.

In short, this article is different from the above surveys. It is the first comprehensive review of real-time SAR ship detection.

### III. MODEL COMPRESSION AND ACCELERATION METHODS

CNN-based deep-learning models have shown great advantage in computer vision tasks especially object detection. But CNN is heavy in computational cost and memory storage, which hinders the usage on some real-time applications, for example, processing on the edge (airplane and satellite). So a lot of methods have been proposed to compress and accelerate the CNN models. They can be divided into six categories: parameter pruning, quantization, knowledge distillation, low-rank factorization, lightweight networks, and model deployment, as shown in Fig. 3. The above methods can be used alone or in combination.

In modern CNN, the fully connected layers gradually disappear, and the convolution layer occupies most of its storage and

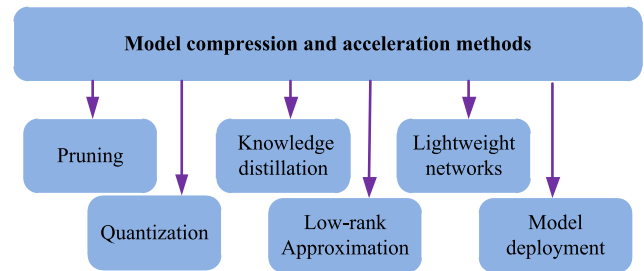


Fig. 3. Model compression and acceleration methods.

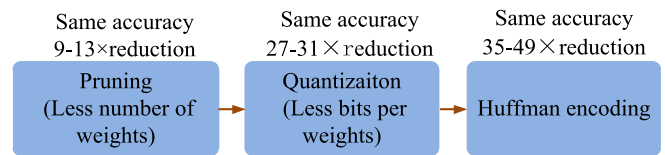


Fig. 4. Three-stage compression method proposed in [94]: pruning, quantization, and encoding.

calculation. Therefore, the main objectives of the above model compression and acceleration methods are the convolution layer. It will take several steps to achieve the goal generally. For example, Han et al. [94] promoted a method to compress CNNs with three steps: pruning, quantization, and Huffman encoding. By this, AlexNet could be compressed by 35 without drops in accuracy, as shown in Fig. 4.

#### A. Pruning

Generally, the CNNs after training are overparameterized, and they have millions or even billions of parameter. For example, ResNet-50 needs 95 MB memory for storage and over 3.8 billion floating number multiplications when processing an image.

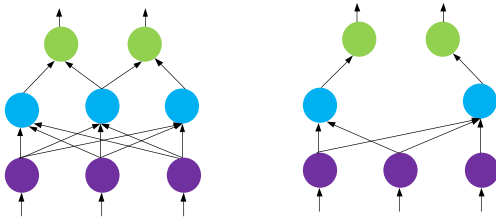


Fig. 5. Results of weight pruning and neuron pruning.

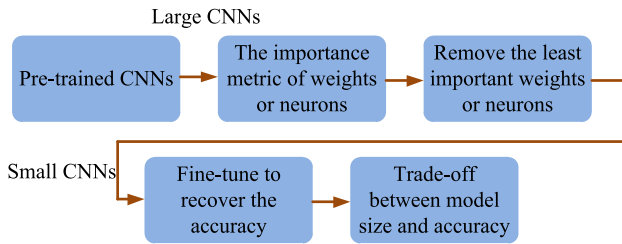


Fig. 6. Process of parameter pruning.

There are significant redundant weights or neurons, which are unimportant or unnecessary [95], [96], [97]. So in order to compress the CNNs, we can prune the weights and neurons that are less important. After pruning, we can get a smaller network, and the CNNs still work as usual but save more parameters and computational time, as shown in Fig. 5. The importance metric of a weight can be L1 norm or L2 norm. The importance metric of a neuron can be the number of times it was not zero after training. After sorting the importance of all weights and neurons, removing unimportant weights or neurons, we can get a smaller network.

Generally, the classification or the detection result will decrease after the pruning. So we should retrain the CNN model to promote the accuracy further. In order to get a good performance on accuracy, we should not prune too much at once [98]. We should gradually prune them as shown in Fig. 6.

Pruning will increase the sparsity of parameters, thereby reducing storage and computing. According to different pruning objectives, pruning can be divided into fine-grained pruning, vector-level pruning, kernel-level pruning, group-level pruning, and filter-level pruning [99].

Fine-grained pruning methods remove parameters with unstructured [100]. The parameters of CNNs can be pruned thoroughly, and are very sparse. These methods have a better compression ratio. But what should be noticed is that as the CNNs are not structural, and they have a lot of fragmented operations which are hard to accelerate on the hardware, e.g., GPU. In other words, although we get fewer parameters, the calculation speed will not be accelerated. So we should pay more attention to the structured pruning. Vector-level methods prune vectors in the convolutional kernels. Kernel-level pruning methods prune convolutional kernels in the filters. They are seldom used, as most pruning methods mainly focus on fine-grained pruning or filter-level pruning. Group-level pruning methods prune the parameters according to the sparse pattern on the filters. By this, convolutions can be implemented by thinned dense matrices

multiplication. So the BLAS can be utilized to achieve a higher speed-up. Filter-level pruning methods prune the filters or the channels. It can make CNNs much lighter. After filter-level pruning, the input channel of the next layer is also smaller. Filter-level pruning methods are structured with less fragmented operations and are easy to be processed by hardware (CPU or GPU). So filter-level pruning methods are the most efficient for compressing CNNs.

The processing of pruning is very trivial as CNNs should be pruned and trained several times. The fine-grained pruning should be used on custom hardware or special data structures for sparse matrices. But the filter-level pruning is not needed and can be used on general processors. Pruning is usually with quantization and encoding for further accelerating CNNs.

### B. Quantization

Network quantization compresses CNNs with less number of bits to represent each weight. It can significantly reduce the memory and computation with less loss on accuracy [101], [102], [103], [104].

Generally, the weights in CNNs are saved as 32-bit floating-point numbers. But for the accuracy of deep-learning task, the number of bits has a weak impact on it. So we can use fewer bits to represent the weights. This process is called the quantization. It can reduce the computation and memory size. For example, if we use 16 bits to store a parameter rather than 32 bits, the model size of the network can be reduced half. The weights can be quantized to 16-bit, 8-bit, 4-bit, or even with 1-bit [105]. The quantization can be also used on gradient and activation. The gradients quantization can accelerate the training stage. The weight and activation quantization methods can accelerate the inference stage.

We can create the clusters of the weights, and all the weights which fall into that cluster can share the same weight value. When storing the network, the real values are not needed to be saved. We only need to record the class ID of the weights and use the means of the weights in the classes to represent the real values of the weights [106]. It can compress the network, but decrease the performance.

The parameters can be represented by a codebook and a set of quantization codes [107]. It represents frequent clusters by fewer bits, and represents rare clusters by more bits. For example, Huffman encoding. It represents the common tokens with less bits, and represents the rare tokens with more bits. By this, the network can be compressed further. It can achieve 4–6 speed-up and 15–20 compression ratio with little accuracy loss.

These methods can also be categorized into two main groups: quantization after training and quantization when training. The former is used to reduce the inference time and save energy. And the latter is used to reduce the network size and make the training process more computationally efficient.

The quantization can use shift or XNOR rather than multiply-accumulate operations in custom hardware [108]. And thus

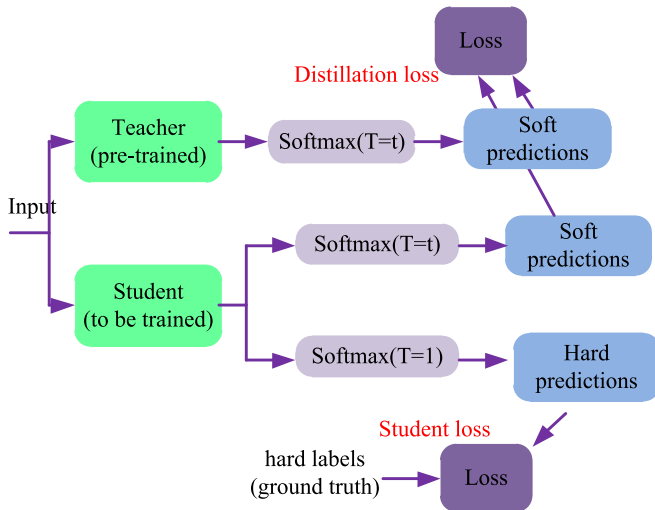


Fig. 7. Principle of knowledge distillation.

reduce the energy consumption. But the CNNs need to be trained several times, which make the training process tedious.

### C. Knowledge Distillation

Hinton put forward the concept of knowledge distillation for the first time in distilling the knowledge in a neural network. And he introduced the soft targets of teacher to induce the training of students' network. The knowledge distillation is classified into three categories, they are logits transfer, teacher assistant, and domain adaptation [109].

Generally speaking, the teacher model has strong ability and performance, while the student model is compact. The knowledge distillation methods transfer the generalization ability of the teacher model to the compact student model to improve its performance with less complexity. The basic idea of knowledge distillation is to transfer the dark knowledge in the complex teacher model to the simple student model. These methods match or outperform the teacher's performance, while requiring notably fewer parameters and multiplications [110], [111], [112], as shown in Fig. 7.

The parameter  $T$  represents temperature. Generally,  $T$  is 1. When  $T$  is larger, a softer probability distribution will be obtained. There are two loss functions. The first loss function requires that the student model and the teacher model use the same  $T$  when calculating the softmax layer. The second loss function requires the student model  $T$  to be taken as 1, and the loss function is the weighted average of the two objective functions. Soft prediction carries more and more useful information than hard prediction. The knowledge distillation can get a lightweight CNN model with high accuracy [113].

The softmax function is formulated as follows:

$$q_i = \frac{\exp(z_i/T)}{\sum_j \exp(z_j/T)}. \quad (1)$$

### D. Low-Rank Factorization

Low-rank factorization is a straightforward way for model compression and acceleration. It is based on the fact that the weight vector is mainly distributed in some low rank subspaces, and a few bases can be used to reconstruct the weight matrix. The low-rank decomposition factorizes multidimensional tensors (in convolutional and fully connected layers) into smaller matrices to eliminate redundant computation. For examples, we can decompose the  $K$  convolutions into two separable convolutions of size  $1 \times K$  and  $K \times 1$ . By this, we can remove redundancy and reduce weight parameters. Low-rank factorization can reduce the computation costs in CNNs. It can be used in both convolutional layers and fully connected layers. It has only a small accuracy loss [114].

Convolution operations contribute most of the computations in CNNs, so we can make the inference process faster by decomposing the convolution layer. A convolutional kernel can be represented as a 4-D tensor  $w \times h \times c \times n$  tensor. They are kernel width, kernel height, and the number of input and output channels, respectively. Ideas based on tensor decomposition is derived by the intuition that there is a significant amount of redundancy in the 4-D tensor, which is a particularly promising way to remove the redundancy. Based on how many components the filters are decomposed into, the low-rank method can be divided into three categories: two-component decomposition, three-component decomposition, and four-component decomposition [115].

The fully connected layers can be view as a 2-D matrix, and it contains around 89% of the parameters in CNNs like AlexNet. Low-rank factorization can also be applied to the fully connected layer. It can make the model storage-friendly [116].

Singular value decomposition (SVD) is a common and popular factorization scheme for reducing the number of parameters. Besides, canonical polyadic decomposition, batch normalization decomposition, Tucker-2 decomposition, and the SVD are the usually used matrix factorization techniques [117].

Low-rank factorization is taken layer by layer. After factorizing the parameters of one layer, subsequent layers are then factorized based on some reconstruction error. It is difficult to be used on deep CNNs, as with the increase of depth, the search space of the decomposition hyper-parameters is huge.

### E. Lightweight Networks

Since AlexNet won the first place in 2012 ILSVRC, deep CNNs encounter another revival. A lot of CNNs have been proposed with excellent performance, for example, VGGNet [118], Inception [119], ResNet [120], ResNeXt [121], and DenseNet [122]. These CNNs mainly consider accuracy but not speed. So they are computation extensive and are very heavy in model size. In order to deploy them on edge devices, a lot of ideas about designing lightweight CNNs have been proposed as shown in Fig. 8. The development of lightweight CNN also promotes the progress of real-time object detection.

1) *SqueezeNet Series*: SqueezeNet [123] has 98% fewer parameters than AlexNet and achieves the same accuracy on the ImageNet dataset. It uses three strategies to reduce the number

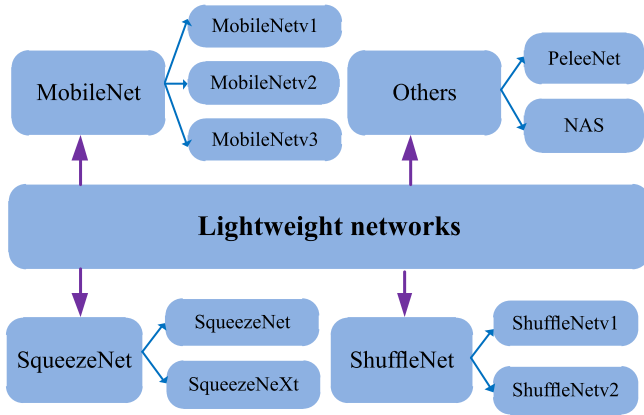


Fig. 8. Lightweight networks.

of parameters. It proposes a new network architecture called fire module. It replaces  $3 \times 3$  convolution with  $1 \times 1$  convolution, and reduces the channel of the input of  $3 \times 3$  convolution. Downsampling is carried out in the later stage of the network, so that the convolution layer has a larger activation feature map. SqueezeNext [124] uses a two-stage squeeze operation to achieve a significant reduction in channels. With separable  $3 \times 3$  convolution, it further reduces model size. By using element by element addition similar to ResNet, a deeper network can be trained without the problem of gradient disappearance. SqueezeNext has fewer parameters and higher accuracy than SqueezeNet.

2) *MobileNet Series*: MobileNetv1 [125] decomposes the standard convolution into 2-D and 1-D convolutions, which effectively reduces the amount of calculation and model parameters. MobileNetv1 uses separable convolution to compress the feature map, which significantly reduces the amount of parameters and significantly improves the speed.

Depthwise convolution converts  $N \times H \times W \times C$  into  $C$  groups, and then each group is calculated by  $3 \times 3$  convolution, which is equivalent to collecting the spatial characteristics of each channel. Pointwise convolution does  $k$  ordinary  $1 \times 1$  convolutions to  $N \times H \times W \times C$ , which is equivalent to collecting the features of each point.

MobileNetv2 [126] was proposed by Google in 2018. The model reduces the amount of computation and memory usage while maintaining accuracy. It has three characteristics: reverse residual, linear bottleneck, and depthwise convolution. MobileNetv2 is spindle type, with large middle and small sides. This is because MobileNetv2 uses  $1 \times 1$  convolution first to increase the number of channels, and then  $3 \times 3$  depthwise convolution can reduce the amount of calculation. Although there are many intermediate channels, the depthwise convolution calculation is small. MobileNetv2 also removes the last ReLU of the bottleneck layer. MobileNetv2 model is compact, with small amount of computation and good classification performance. It shows good performance in detection and segmentation tasks.

MobileNetv3 [127] is improved on MobileNetv2, it explores how neural architecture search (NAS) and manual design work together and complement each other. It first uses MnasNet [128]

to search the rough structure, then uses reinforcement learning to select the optimal configuration from a set of discrete choices, and then uses MnasNet to fine tune the architecture.

3) *ShuffleNet*: Although the pointwise convolution reduces the number of parameters, there is a problem of low computational efficiency, because a large number of  $1 \times 1$  convolution will consume a lot of computing resources. ShuffleNet uses pointwise group convolution to reduce the amount of computation, but there is no connection between groups, which will affect the performance of the network. Therefore, channel rearrangement is used to strengthen the connection between different groups.

Due to the addition of pointwise grouping convolution and channel rearrangement, the calculation of ShuffleNet is more efficient. Compared with ResNet and ResNeXt, the computational complexity of ShuffleNet is the smallest, for example, considering the input size of  $c \times h \times w$ . Bottleneck channel is  $m$ , ResNet unit needs  $h \times w \times (2 \times c \times m + 9 \times m \times m)$  FLOPs, ResNeXt unit requires  $h \times w \times (2 \times c \times m + 9m \times m/g)$  FLOPs, but the ShuffleNet unit only needs  $h \times w \times (2c \times m/g + 9 \times m)$  FLOPs, where  $g$  represents the number of groups of convolution.

ShuffleNetv1 [129] uses pointwise group convolution and channel shuffle to greatly reduce the amount of calculation and ensure the accuracy. It has a lower error rate than MobileNet (the top-1 error rate is 7.8%), and is 13 times faster than AlexNet on the ARM chip of mobile devices.

ShuffleNetv2 [130] proposes that the following two factors must be considered in the structural design of CNN: first, the direct measurement standard (speed), rather than the indirect measurement standard (flop); second, measurement needs to be carried out on the target platform. The article gives four practical guiding principles: principle 1: the same number of input and output channels can minimize memory access; principle 2: excessive use of group convolution will increase memory access cost (MAC); principle 3: network fragmentation will reduce parallelism; and principle 4: element level operations cannot be ignored. According to the guiding principles, the author designed ShuffleNetv2, which achieved a good tradeoff in speed and accuracy.

4) *Others*: PeeleNet [131] used two-way dense layers, stem block, dynamic number of channels in a bottleneck, transition layer compression, and conventional post activation to reduce computation cost and increase speed. The stem block was used to alleviate information loss. PeeleNet consists of a stem block, four stages of modified dense and transition layers, and ultimately the classification layer.

The CNN structure mentioned above is basically designed manually by experience, and these networks are not necessarily optimal. NAS [132] can find lightweight network models. It can use search strategy to automatically find the optimal CNN structure in the search space. Mnasnet is an automated NAS approach. It formulates the search problem as multiobject optimization aimed at both high accuracy and low latency. MnasNet was almost twice as fast as MobileNetv2 while having better accuracy.

The above lightweight CNNs relied on depthwise separable convolution, which lacked efficient implementation on most

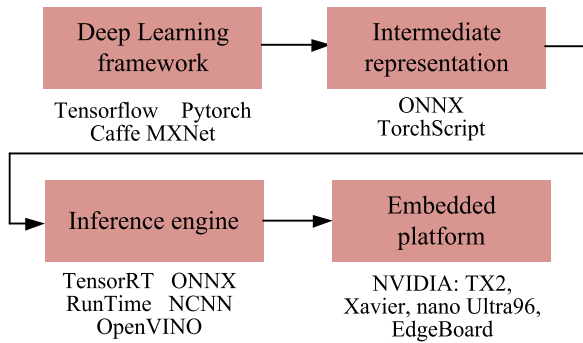


Fig. 9. Common process of model deployment.

hardware. It should be noted that when these CNN are deployed on hardware devices, some models seem lightweight, but the hardware does not support some calculations, resulting in poor real-time performance.

#### F. Model Deployment

Computational acceleration is also a method to realize real time. It includes using FFT-based convolutions and fast convolution using the Winograd transformation. Winograd transformation is another equivalent method for convolution, which improves the calculation speed. Convolution in the time domain is equivalent to pointwise multiply in the frequency domain.

The key to realize real-time object detection is to deploy lightweight CNN models on the customized hardware. Model deployment refers to deploying the training model generated by deep learning to various cloud, edge, and edge devices to make it run efficiently, so as to apply the algorithm model to various tasks in reality. The common process of model deployment is shown in Fig. 9. It includes four steps: deep-learning framework, intermediate representation, inference engine, and embed platform.

The deep-learning framework is used to define the network structure and determine the parameters in the network through training. Tensorflow [133], Pytorch, and Caffe [134] are the usually used deep-learning framework.

The intermediate representation can solve the problem of neural network model transformation between different training and inference frameworks. It only describes the structure and parameters of the network. ONNX (open neural network exchange) is an intermediate representation [135]. It defines a set of extensible calculation diagrams and a series of standard data types and operators. Currently, most training, deployment frameworks, and reasoning acceleration engines of hardware manufacturers support ONNX format.

The inference engine written codes with high-performance programming frameworks (such as CUDA and OpenCL), it can efficiently execute operators in deep-learning networks. The ONNX, RunTime, TensorRT, MNN, NCNN, and OpenVino are the inference engine that we can use [136], [137].

The embedded platform is used to realize CNN's real-time processing under the condition of limited power consumption and computing power. The NVIDIA Jetson TX2, Xilinx Ultra96



Fig. 10. Software and hardware codesign.

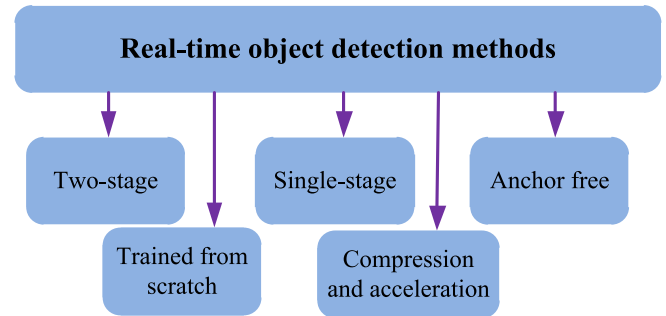


Fig. 11. Real-time object detection.

with UltraScale and ZU3, Huawei Atlas 200 with Hi3559, and Baidu EdgeBoard with FPGA are the embedded platform we can use [138], [139].

The best solution for realizing real-time processing is algorithms and hardware codesign as shown in Fig. 10. The algorithms for efficient inference include pruning, quantization, low-rank factorization, winograd transformation, and so on. The hardware should be efficient in inference. It should minimize memory access. It needs to support some lightweight operations. The hardware also needs to optimize some special operations, for example,  $1 \times 1$  convolution,  $3 \times 3$  convolution, group convolution, and depthwise convolution.

#### IV. REAL-TIME OBJECT DETECTION METHODS

The methods to realize real-time object detection can be summarized as Fig. 11. They are two-stage, single-stage, anchor free, trained from scratch, and the compression and acceleration.

According to the number of stage, the deep-learning-based object detection algorithm can be divided into single-stage detectors and two-stage detectors. During the development of object detection, both of the single-stage detectors and two-stage detectors are continued to pursue speed and accuracy. For example, from R-CNN [140], SPP-Net [141], and Faster R-CNN [142] to R-FCN [143] and Light-head R-CNN [144], the detectors are becoming faster and more accurate. More than this, the YOLOv1 [145], YOLOv2 [146], YOLOv3 [147], YOLOv4 [148], YOLOv5 [149], and YOLOX [150] also reflect this phenomenon. We will introduce it in detail below.

The anchor-free detectors have been widely studied and a large number of achievements have emerged recently. The anchor-free detectors show great potential in real-time SAR ship detection. This is because ships in SAR images are very spare, and most of the anchors are invalid in the anchor-based detectors. Besides this, the trained from scratch technique also show great advantages in realizing lightweight detectors. This is because SAR images are different from optical image of ImageNet, the

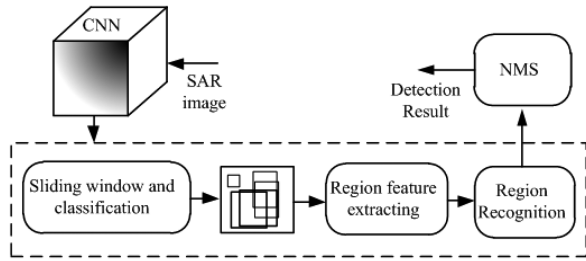


Fig. 12. Principle of two-stage detectors.

Selective search for generating candidate region	Selective search for generating candidate region	CNN is responsible for generating candidate region, extracting features, classification and regression
CNN is responsible For extracting features	CNN is responsible for extracting features, classification and regression	
classification (SVM)		
regression		
66% mAP 47s/image	70% mAP 3s/image	73.2% mAP 0.2s/image
R-CNN	Fast R-CNN	Faster R-CNN

Fig. 13. Evolution of R-CNN, fast R-CNN, and faster R-CNN.

loaded pretrained parameters are not suitable, and are redundant for SAR ship detection.

Besides the above ways, the compression and acceleration of CNNs (pruning, quantization, and knowledge distillation) are also used on the object detection method.

#### A. Two-Stage Real-Time Detectors

The principles of two-stage detection algorithms are shown in Fig. 12. The two-stage detectors use a CNN to classify and regress these anchor boxes twice to obtain the detection results.

Classical two-stage detectors are faster R-CNN, R-FCN, feature pyramid networks (FPN) [151], cascade R-CNN [152], mask R-CNN [153], and so on [154]. Faster R-CNN is the foundation work, and most of the two-stage detectors are improved based on it.

The evolution process of R-CNN, fast R-CNN, and faster R-CNN are shown in Fig. 13. We can see that CNN not only is used for extracting features but also can be used for generating candidate region.

R-CNN uses the selective search algorithm to generate the candidate box of the object, and inputs it as a sample into the CNN. The CNN generates the positive and negative sample features, and forms the corresponding feature vector. Then, the support vector machine classifies the feature vector. After the regression, the category and location are output. R-CNN gets mAP of 53.3%, which is 13.4% higher than the best traditional detector. R-CNN shows the great advantage of deep learning.

SPP-Net proposed spatial pyramid pooling (SPP) to solve the problem of feature extraction in R-CNN. SPP-Net can get fixed length feature vectors through CNN no matter how length is the input.

Fast R-CNN adopts the idea of multitask loss function. The classification loss and regression loss are unified to train. It no longer requires additional hard disk space to store the middle layer features. And the gradient can be directly propagated through the ROI pooling layer. Fast R-CNN processes one image

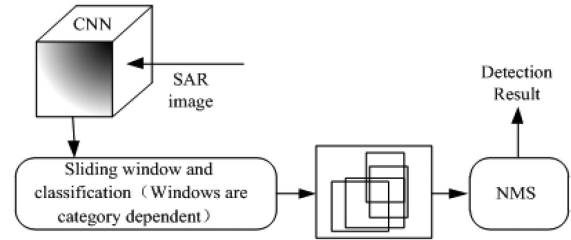


Fig. 14. Principle of single-stage detectors.

with 2–3 s, and the main time is spent on candidate region extraction.

Faster R-CNN adopts the shared CNN to predict the region proposal. It includes regional proposal network (RPN) and fast R-CNN. RPN is used to generate candidate windows. It uses the anchor box mechanism, which greatly reduces the amount of computation by directly generating candidate windows on the feature map. Faster R-CNN slightly improves the accuracy, and greatly improves the speed [17 frames per second (FPS)]. It can be processed in an end-to-end way. GPU can be used to accelerate the calculation throughout the whole process.

Fast R-CNN and faster R-CNN apply a per-region subnetwork many times to classify and regress the targets, which is time consuming. R-FCN uses a fully convolutional structure and all the computation shared on the entire image. It is more accurate and efficient. As deeper layers in the convolutional network are translation-invariant, making them ineffective for localization tasks. R-FCN proposed the position-sensitive score maps to solve this problem. It uses position-sensitive score maps to address the dilemma between translation-invariance in image classification and translation-variance in object detection. R-FCN achieves 83.6% mAP on the PASCAL VOC 2007 with ResNet-101 as the backbone.

Faster R-CNN has two fully connected layers for ROI recognition, and R-FCN has a large score maps. They both perform an intensive computation after or before ROI warping. Though the backbone is lightweight, the above two-stage detectors are still slow. Light-head R-CNN solves this shortcoming of faster R-CNN and R-FCN. The head of light-head R-CNN uses a thin feature map and a cheap R-CNN subnet, which makes the head of the detector as light as possible. It gets 30.7 mAP at 102 FPS on COCO.

From R-CNN, SPP-Net, fast R-CNN, and faster R-CNN to R-FCN and light-head R-CNN, we can find that the two-stage detectors always pursue the improvement of accuracy and speed by sharing features on the backbone or the head layers. But as the two-stage detectors accurately locate and classify targets through two stages, which increase the amount of computation. So it is inherently not suitable for real-time object detection. Researches usually seek solutions in single-stage detector when designing real-time object detection.

#### B. Single-Stage Real-Time Detectors

The principles of single-stage detection algorithms are shown in Fig. 14. It uses a full convolution network to classify and



TABLE II  
YOLO SERIES OBJECT DETECTOR

YOLO	Backbone	Neck	Head	Prediction
YOLOv1	GoogLeNet	None	FC->7×7×(2×5+20)	MSE loss
YOLOv2	DarkNet19	None	Conv->13×13×5×(5+20)	MSE loss
YOLOv3	DarkNet53+ CSP	FPN	Conv->13×13×5×(5+80) ->26×26×5×(5+80) ->52×52×5×(5+80)	MSE loss
YOLOv4	DarkNet53+ CSP	FPN+P AN+SP P	Conv->13×13×5×(5+80) ->26×26×5×(5+80) ->52×52×5×(5+80)	DIoU Loss
YOLOv5	DarkNet53+ Focus+CSP	FPN+P AN+SP P+CSP	Conv->13×13×5×(5+80) ->26×26×5×(5+80) ->52×52×5×(5+80) Conv->20×20×5×(5+80)	CIoU Loss
YOLOX	DarkNet53	FPN	->40×40×5×(5+80) ->80×80×5×(5+80) Decoupled head	Anchor free SimOTA IoU Loss

regress the boxes once to get the detection results. The single-stage detectors only need to look at the picture once, and can predict what the object is and where the object is. It is similar to human eyes. So they are faster than two-stage detectors. Classical single-stage detectors are YOLO, SSD [155], RetinaNet [156], and CornerNet [157]. YOLO is most popular single-stage detection algorithms, and most of the subsequent single-stage works are based on them.

Among the single-stage detectors, YOLO series are popular in real-time applications due to excellent speed and accuracy tradeoff. The structures of YOLO series object detector are shown in Table II.

YOLOv1 regards object detection as a regression problem, and it outputs the spatially separated bounding box and related class probability simultaneously. YOLOv1 divides the input picture into  $S \times S$  grids, each grid cell predicts  $B$  boxes and the confidence scores corresponding to these boxes. For PASCAL VOC,  $S = 7$ ,  $B = 2$ ,  $C = 20$ . The final prediction output is a  $7 \times 7 \times 30(5 \times 2 + 20)$  tensor.

YOLOv2 uses the multiscale training method. It predicts the offset rather than the parameter itself. It uses an anchor mechanism to obtain anchor box parameters by clustering the object size in the dataset. And every cell predicts five anchor boxes. The backbone network is DarkNet-19. The detection head has changed from  $7 \times 7$  to  $13 \times 13$ . Batch normalization, pass-through, high resolution, and multiscale training are used to promote the performance further.

YOLOv3 uses DarkNet-53 as the backbone. DarkNet-53 reduces the output feature map to 1/32 of the input, which is stronger than DarkNet-19 and more efficient than ResNet-101 and ResNet-152. The predictions are done on three different branches:  $13 \times 13$ ,  $26 \times 26$ , and  $52 \times 52$ . The anchor mechanism of YOLOv3 is the same as that of YOLOv2. YOLOv3 can process an image of  $320 \times 320$  in 22 ms.

YOLOv4 consists of CSPDarknet-53 backbone, SPP + path aggregation network (PAN) based neck, and head of YOLOv3. It uses two anchors for one ground truth, while YOLOv3 uses only one anchor for one ground truth. It also uses several techniques to achieve state-of-the-art results. It uses bag-of-freebies to improve the performance without increasing the inference

time. For example, cutMix and mosaic data augmentation, dropBlock regularization, class label smoothing, Ciou-loss, self-adversarial training, and so on. It uses bag-of-specials to improve the performance, and only a small amount of calculation is increased. For example, mish activation, cross-stage partial (CSP) connections, SPP, SAM (spatial attention module), PAN, Diou-nonmaximum suppression (NMS), and so on. YOLOv4 gets 43.5 mAP with 65 FPS on MS COCO.

YOLOv5 adopts adaptive anchors and uses the network to learn anchor parameters. Its backbone is based on DarkNet53 with focus and CSP. In the neck part, the structure of an FPN and PAN are adopted. Its prediction head is the same as YOLOv3 and YOLOv4. YOLOv5 is a state-of-the-art object detection algorithm with fast inference speed and exact accuracy.

YOLOX introduces advanced anchor-free method to improve the performance of detector, significantly outperforming YOLOv5 in terms of precision. YOLOX uses a decoupled head to generate two-way feature maps by two separate  $1 \times 1$  convolutional layers. The decoupled head improves the performance of the detector. The SimOTA is used to dynamic assign label. YOLOX achieve 50.0% AP on MS COCO with 68.9 FPS on Tesla V100, which surpasses YOLOv5-L by 1.8% AP. And the numbers of parameters of them are roughly the same.

For the YOLO series algorithm, other researchers have proposed some improvements, and the speed has increased significantly, for example, PP-YOLO [158], PP-YOLOv2 [159], and PP-PicoDet [160].

YOLO series show great potential for real-time object detection. And a lot of researchers apply them into SAR ship detection and get good results. We will review them in Section V.

### C. Anchor-Free Detectors

Deep-learning-based object detection algorithms can be divided into anchor-based and anchor-free-based. The anchor-based algorithm uses the anchor box as a reference to search the region that may contain the object. These scale and aspect ratio of the anchor box are designed according to the statistical of the dataset. The anchor-based detectors make great contribution to the development of deep-learning-based object detection, for example, faster R-CNN, YOLOv2, YOLOv3, and SSD. However, it is discrete sampling and needs to be designed manually by experience, so there will be the following disadvantages [161]:

- 1) The anchor box needs to be “carefully” designed according to different datasets. This will introduce many hyper-parameters that need to be optimized, such as the number, size, and aspect ratio of the anchor box. If the parameters change, the detection performance will fluctuate.
- 2) In order to obtain a good recall rate, it will generate a large number of anchor boxes. For example, SSD has 8732 anchor boxes, and few can match the ground truth, most of them are invalid, so there is a large redundancy.
- 3) There are fewer positive samples and most of them are simple negative samples. This imbalance between positive and negative samples will reduce the performance of the model.

- 4) When the dataset changes, it is necessary to change the anchor box according to the size and shape of the targets in dataset.

The anchor-free detector opens up another idea by eliminating the predefined anchor box. It can directly predict several key points of the target from the feature map. These algorithms are exploring how to efficiently use points to represent a bounding box. CornerNet uses the upper left and lower right points of the box to represent the box. FCOS [162] and FoveaBox [163] do the detection process by pixel prediction.

The anchor-free detectors can avoid various problems, and has great application potential in SAR ship detection. For example, due to the small size and sparse distribution of ships, most of the candidate anchor boxes are invalid negative samples. The anchor-free detectors can neglect the invalid anchors and reduce the amount of the predicted boxes, and thus improve the accuracy and speed simultaneously.

For the special scene of SAR ship detection, due to the small size and sparse distribution of ships, most of the candidate anchor boxes are invalid negative samples. The anchor-free detector avoids various problems in anchor box generation. So it has great application potential here.

#### D. Trained From Scratch Detectors

Since Ross proposed R-CNN, object detection algorithms need to load the weight pretrained on classification dataset (ImageNet) and fine-tune parameters to adapt to the new detection task. This transfer learning can make the detection algorithm initialize better and make up for the problem of insufficient samples. But there will be the following problems [164]:

- 1) The loss function and category distribution between classification and detection are different, so the transferred parameters are not suitable for detection, which will make the detection algorithm less than optimal.
- 2) The detection includes two subtasks: classification and localization, which are optimized at the same time. However, these two tasks are contradictory in nature. For classification, translation invariance is required, but for detection, translation invariance is required. It is unreasonable to use the pretrained model for object detection.
- 3) Most networks will produce high receptive fields through multiple downsampling in the latter layers, which is good for classification. However, this will sacrifice the spatial resolution of the feature map, and it is difficult to accurately locate object.
- 4) Most detection algorithms directly borrow the model structure and parameters after pretraining and cannot modify the structure. It hinders researchers from designing CNN flexibly according to their needs.
- 5) At present, the network is generally designed for three channel natural images. For single-channel SAR images, there are too many channels and parameters are redundancy.

In order to solve the problems of transfer learning, algorithms trained from scratch are proposed, for example, DSOD, DetNet [166], ScratchDet [167], and so on [168].

DSOD and GRP-DSOD [165] realize training from scratch through well-designed backbone network and frontend network. The parameters of the detection algorithm are greatly reduced, and the accuracy is equivalent to the most advanced detection algorithm at that time. It summarizes four principles for designing backbone networks: single-stage, dense prediction structure, stem unit, and deep supervision.

DetNet designs a backbone network for detection tasks. Considering the contradiction between detection and classification tasks, the backbone network retains a larger scale in the last few layers, which can retain more location information.

ScratchDet proposed that increasing the learning rate while using BN in each layer can make the detection algorithm more robust and converge faster. At the same time, based on ResNet-18, the article proposes a backbone network root Root-ResNet for detection algorithm, which uses three stacked  $3 \times 3$  convolution instead of  $7 \times 7$  convolution, and removes the max-pool layer at the front to reduce the information loss.

He et al. [168] uses group normalization and asynchronous BN to increase the batch size, and the direction of gradient descent is more accurate. So it can accelerate convergence and improve the accuracy of convergence.

The model trained from scratch not only has high accuracy, but also greatly reduces the size and amount of calculation of the model. Due to the above advantages, it is also used in SAR ship detection.

#### E. Compression and Acceleration Methods

To detect objects on platforms with limited computing power and memory resources, researchers also used CNN's compression and acceleration methods for object detection. For example, pruning, quantization, knowledge distillation, and low-rank factorization are usually in the inference engine, such as TensorRT, ONNX RunTime, NCNN, and OpenVINO. Generally, the knowledge distillation, pruning, and quantization are used from front to back with multiple times. Among them, the knowledge distillation is usually used in real-time object detection [169], [170], [171], [172], [173].

### V. REAL-TIME SAR SHIP DETECTION

As far as we know, there are 70 public papers those are about real-time SAR ship detection. We divide them into the following seven categories as shown in Fig. 15. They are anchor free, trained from scratch, YOLO series, CFAR+CNN, lightweight backbone network, model compression and acceleration, and hardware deployment. The datasets and evaluation metrics used in this area are also reviewed in this part.

#### A. Seventy Public Papers About Real-Time SAR Ship Detection

Table III shows the number of papers in each year. From the table, we can see that 2019 is the beginning year of real-time SAR ship detection. It is behind of the usage of deep learning in

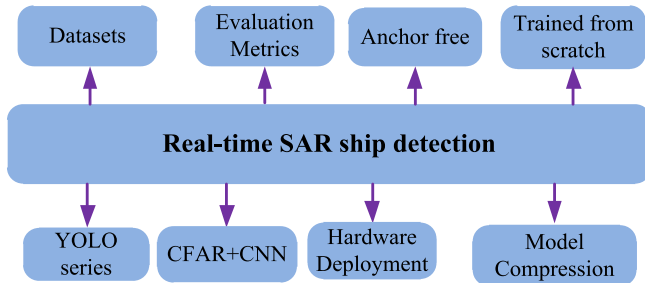


Fig. 15. Real-time SAR ship detection.

TABLE III  
NUMBER OF PAPERS IN EACH YEAR

Years	2019	2020	2021	2022
Number of paper	5	18	25	22

TABLE IV  
TIMES THAT DATASETS USED

Data sets	SS DD	SAR-Ship-Dataset	AIR-SARShip-1.0/2.0	HR SID	LS-SSD D-V1.0
Times	46	13	5	15	4

TABLE V  
DEEP LEARNING FRAMEWORKS THAT ARE USED

Frameworks	Caffe	Keras	Tensorflow	Pytorch
Times	3	2	7	29

this area in 2017. We can also find that the numbers show a trend of increasing. This shows that more and more people begin to pay their attention on real-time SAR ship detection.

Table IV shows the times that datasets are used. We can find that SSDD is used 46 times among the 70 papers. The reason why SSDD is welcomed are that SSDD is the first public datasets (it is more than 1 year earlier than the second dataset) and are friendly to use. But as the emergency of other big datasets, SSDD shows some drawbacks gradually. And the new datasets are better than SSDD in some extent. SAR-Ship-Dataset and HRSID are also usually used in this area. AIR-SARShip-1.0/2.0 and LS-SSDD-V1.0 are less used; this is partly because they are large SAR images and are hard to use.

From Table IV, we can see that researchers have too much choice when conducting experiments. This is not good for the development of algorithms. What is more, the small datasets are prone to overfit when training. So in the future, it is necessary to merge the several public datasets into a large one.

Among the 70 papers, 18 papers are from conferences and 52 papers are from journals. The IEEE TRANSACTIONS ON GEO-SCIENCE AND REMOTE SENSING (Letters) and *MDPI Remote Sensing* are the two mostly appeared journals.

The deep-learning frameworks that are used are shown in Table V. We can find that Pytorch is the most welcomed framework. This is because it is easy to use in research.

Among the 70 papers, 6 papers trained on GPU and tested on the edge, e.g., NVIDIA Jetson TX2, FPGA, and so on, the other

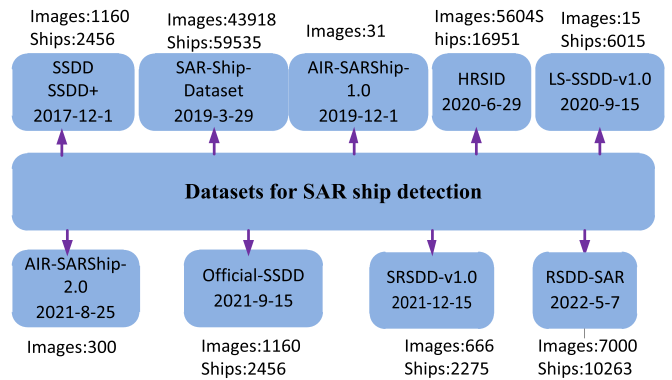


Fig. 16. Ten public datasets in SAR ship detection.

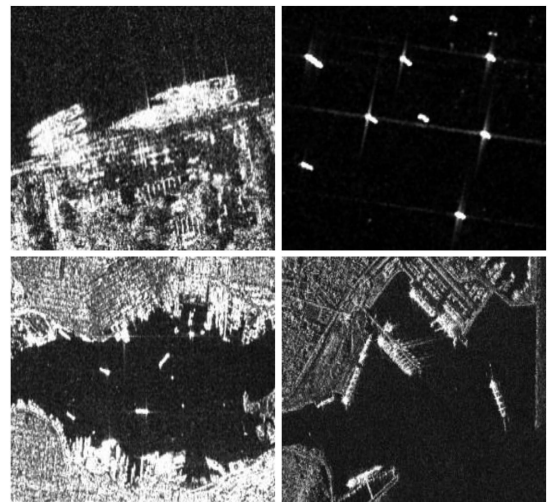


Fig. 17. Samples of SSDD, SSDD+, and official-SSDD.

papers trained and tested on GPU. In the following part, we will survey them in detail.

## B. Datasets

1) *Overview of Public Datasets:* Up to now, there are 10 public datasets in SAR ship detection. They are shown in Fig. 16. They are SSDD (SSDD+), SAR-Ship-Dataset [174], AIR-SARShip1.0 [175], HRSID [176], LS-SSDD-v1.0 [177], AIR-SARShip2.0 [178], Official-SSDD, SRSDDD-v1.0 [179], and RSDD-SAR [180]. SSDD is the first dataset that is used in SAR ship detection. It brings SAR ship detection into deep-learning era. Since then, many researchers have used deep-learning methods to detect ships in SAR images. As the deep-learning methods are data hungry, they need lots of images to train the huge models. SSDD faced the problem of insufficient data volume. So other datasets are proposed successively. The samples of them are shown in Figs. 12–23.

Table VI shows the other information of 10 public datasets. From the table, we can see that most datasets are annotated with vertical bounding box. SSDD+, SRSDDD-v1.0, and RSDD-SAR are annotated with oriented bounding box. HRSID and Official-SSDD are annotated with polygon bounding box.

TABLE VI  
OTHER INFORMATION OF 10 PUBLIC DATASETS

Dataset	Source	Resolution	Image size	No. of Images/ships	annotation
SSDD (SSDD+)	RadarSat-2 TerraSAR Sentinel-1	1 m–15 m	190–668	1160/2456	vertical oriented
SAR-Ship-Dataset	Gaofen-3 Sentinel-1	3 m–25 m	256 × 256	43 918/59 535	vertical
AIR-SARShip-1.0 AIR-SARShip-2.0*	Gaofen-3	1 m, 3 m	3000 × 3000 1000 × 1000	31 300	vertical
HRSID	Sentinel-1 TerraSAR	0.5 m, 1m, 3 m	800 × 800	5604/16 951	polygon
LS-SSDD-v1.0	Sentinel-1	5 m, 20 m	24 000 × 16 000	15/6015	vertical
Official-SSDD	The same as SSDD				polygon
SRSDD-v1.0	Gaofen-3	1m	1024 × 1024	666/2275	oriented
RSDD-SAR	Gaofen-3 TerraSAR	2 m–20 m	512 × 512	7000/10 263	oriented

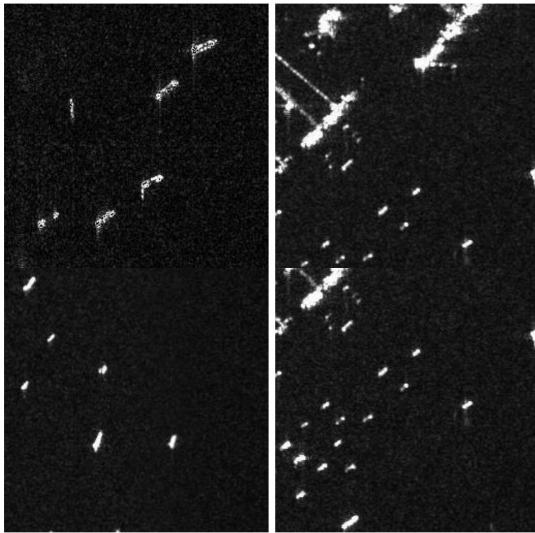


Fig. 18. Samples of SAR-Ship-Dataset.

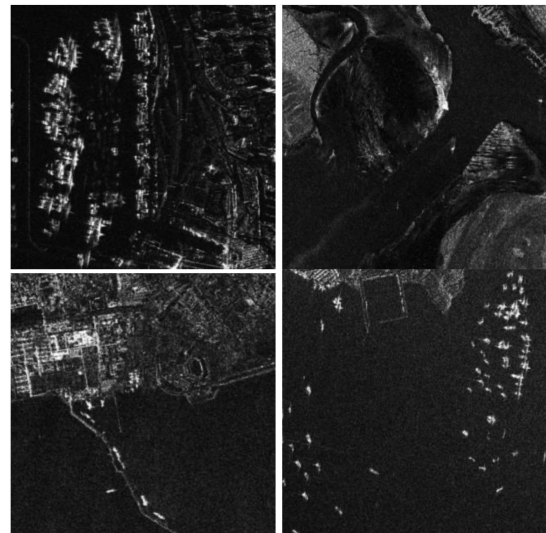


Fig. 19. Samples of AIR-SARShip 1.0.

In the following part, we will introduce the details of the datasets and evaluate the advantages and drawbacks of them.

2) *SSDD*, *SSDD+*, and *Official-SSDD*: *SSDD* is proposed at the conference of 2017BIGSAR DATA in December 1, 2017. It has 1160 images and 2456 ships. The data resources are RadarSat-2, TerraSAR, and Sentinel-1 with resolutions from 1 to 15 m. The length or the width is about 600 pixels. The samples of *SSDD* are diverse, which are helpful for training a robust detector. The length, width, and aspect ratio of the ship bounding box in *SSDD* are counted, which is helpful for designing anchor boxes for the detectors. *SSDD+* is the improved version of *SSDD* with oriented bounding box. *SSDD* and *SSDD+* share the same images.

Due to the insufficient understanding of deep-learning object detection algorithm, there are some problems in *SSDD*, for

example, the coarse annotations and ambiguous standards of use. It hinders fair comparisons and effective academic exchanges in this field. In order to solve this problem, Zhang et al. [85] proposed **Official-SSDD**. It has bounding box, rotatable bounding box, and polygon segmentation. The five using standards are also formulated, they are the training-test division determination, the inshore—offshore protocol, the ship-size reasonable definition, the determination of the densely distributed small ship samples, and the determination of the densely parallel berthing at ports ship samples. **Official-SSDD** is beneficial for fair method comparison and effective academic exchanges in the future.

3) *SAR-Ship-Dataset*: *SAR-Ship-Dataset* is public at March 29, 2019. Its data sources are 102 Chinese Gaofen-3 images and 108 Sentinel-1 images. It has 43 819 chips of 256 × 256 pixels and 59 535 ships with different scales. The resolutions are from 3

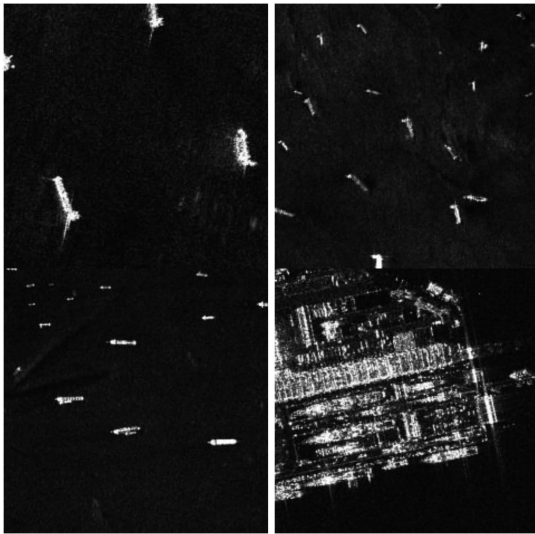


Fig. 20. Samples of HRSID.

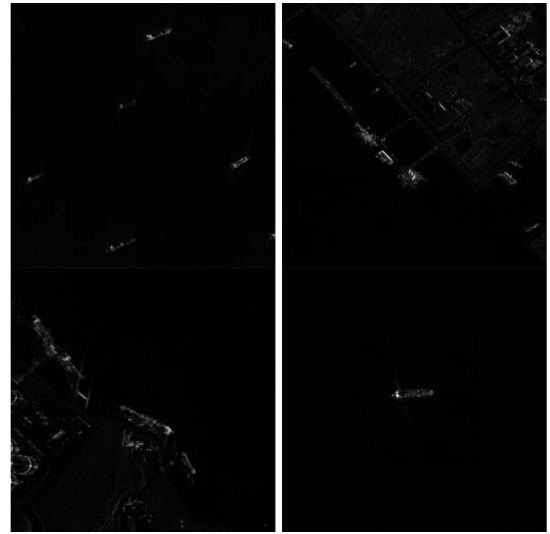


Fig. 22. Samples of SRSDD-v1.0.

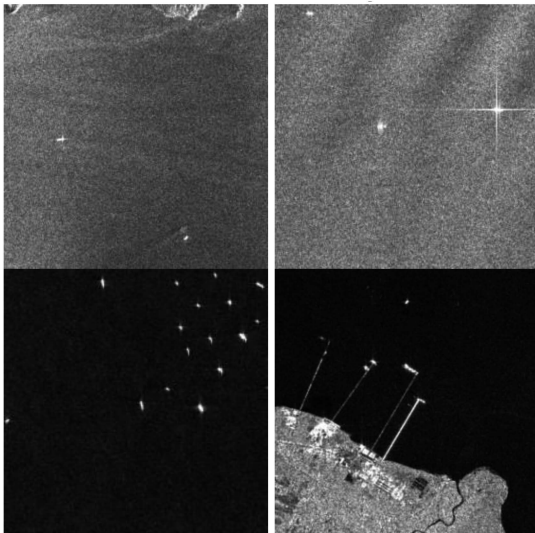


Fig. 21. Samples of LS-SSDD-v1.0.

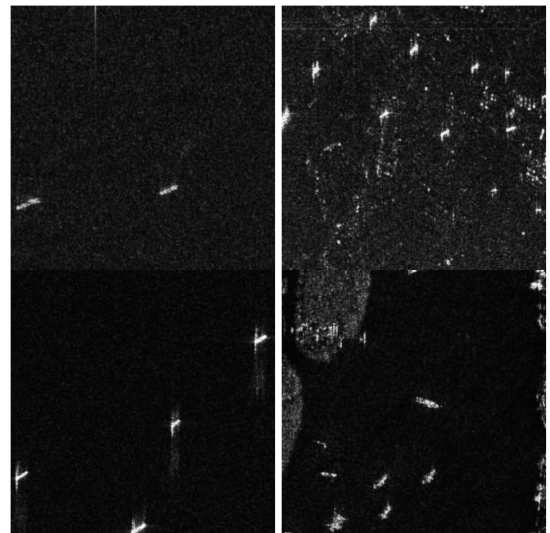


Fig. 23. Samples of RSDD-SAR.

to 25 m. The ship sizes are relatively small and the backgrounds for ships are complex.

4) *AIR-SARShip*: *AIR-SARShip* 1.0 was proposed at December 1, 2019. It has high-resolution and large-scale images. It contains 31 images from Gaofen-3, including harbors, islands, reefs, and the sea surface. The training set has 21 images, and the testing set has 10 images. The image resolutions include 1 and 3 m. The image size is about  $3000 \times 3000$  pixels. *AIR-SARShip*-2.0 is proposed at August 25, 2021. It has 300 images from Gaofen-3. The image size is about  $1000 \times 1000$  pixels.

5) *HRSID*: *HRSID* is proposed at June 29, 2020. *HRSID* includes 99 Sentinel-1B images, 36 TerraSAR-X images, and 1 TanDEM-X image. It has resolutions of 0.5, 1, and 3 m. It is annotated with polygon. *HRSID* can be used in object detection and instance segmentation tasks. *HRSID* has 136 large SAR images with resolution from 1 to 5 m, and they are cropped

to  $800 \times 800$  pixels small images with overlapped ratio of 25%. *HRSID* has 5604 cropped SAR images and 16 951 ships. A total of 65% of the SAR images are training set, and the other 35% are test set. It follows the principle of MS COCO in annotation and scale division. They are 54.5%, 43.5%, and 2% small, medium, and large ships, respectively. The bounding box area of small, medium, and large ships accounts for 0%–0.16%, 0.16%–1.5%, and 1.5% of SAR images, respectively. So *HRSID* has the characteristics of small objects but large scenes; ships are sparsely distributed in SAR images. *HRSID* has the features of small and sparse ships with large scenes.

6) *LS-SSDD-v1.0*: *LS-SSDD-v1.0* is proposed at September 15, 2020. The data source is Sentinel-1 with resolutions of 5 and 20 m. *LS-SSDD-v1.0* can be used in ship detection in large-scene spaceborne SAR images, which can meet the practical application. It has 15 large-scale SAR images, whose ground

		Ground Truth	
		P	N
Prediction	P	True Positives	False Positives
	N	False Negatives	Truth Negatives

Fig. 24. Confusion matrix.

truths are correctly labeled by expert, the automatic identification system, and Google Earth. Each image is divided into 600 subimages with  $800 \times 800$  pixels. The dataset contains 6015 ships. LS-SSDD-v1.0 has the following characteristics: contain large scenes, focus on the small ships, rich pure background, etc. It also provides a large number of performance benchmarks of detection algorithms on datasets.

7) *SRSDD-v1.0*: SRSDD-v1.0 is proposed at December 15, 2021. The data source is Gaofen-3 with 1-m resolution. SRSDD-v1.0 is annotated with oriented bounding box. It contains six categories of ships: ore-oil ships (166), bulk cargo ships (2053), fishing boats (288), law enforcement ships (25), dredger ships (263), and container ships (89). The image size is set to  $1024 \times 1024$ . The annotation format is the same as DOTA. The coordinates of the four corners of the box, the category, whether it is difficult to identify is given on annotation files. It contains 666 images. A total of 420 images with 2275 ships include land cover. A total of 246 images with 609 ships only contain sea in the background. It has six categories.

8) *RSDD-SAR*: RSDD-SAR is proposed at July 7, 2021. The data sources are Gaofen-3 and TerraSAR with resolutions of 2–20 m. RSDD-SAR has 84 GF-3 scenes, 41 TerraSAR-X scenes uncropped large images, including 7000 slices and 10 263 ships. It is annotated by automatic annotation and manual correction with oriented bounding box. The angle of ships in the dataset is evenly distributed between  $0^\circ$  and  $180^\circ$ , and the aspect ratio is concentrated between 2 and 6. The training set has 5000 samples and the testing set has 2000 samples. It has a large number of small ships. It contains vast sea areas, ports, docks, waterways, and other scenes with different resolutions, which are suitable for practical applications.

### C. Evaluation Metrics

The confusion matrix is shown in Fig. 24. From this, we can see the concept of TP, FP, FN, and TN. For example, TP means that the ground truth is positive and the prediction is also positive, FP means that the ground truth is positive but the prediction is also negative.

Based on the confusion matrix, the false positives rate and the true positives rate are calculated as follows:

$$\text{FP rate} = \frac{\text{FP}}{N} \quad (2)$$

$$\text{TP rate} = \frac{\text{TP}}{P}. \quad (3)$$

The accuracy is generally used to evaluate the global accuracy of a model. It cannot contain too much information and cannot comprehensively evaluate the performance of a model. It is calculated as follows:

$$\text{accuracy} = \frac{\text{TP} + \text{TN}}{P + N}. \quad (4)$$

Precision represents the proportion of ships that were correctly detected in all positive detected result. It is calculated as follows:

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}. \quad (5)$$

Recall represents the proportion of ships that were correctly detected in the ground truth. It is calculated as follows:

$$\text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \quad (6)$$

Precision and recall are contradictory, that is to say, the higher the recall, the lower the precision is. In order to give consideration to precision and recall at the same time, F1 score is proposed, which is the harmonic average of precision and recall, with the maximum of 1 and the minimum of 0. It is calculated as follows:

$$\text{F1 - score} = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}. \quad (7)$$

PRC (precision recall curve) is also usually used in object detection. The  $x$ -axis and  $y$ -axis of PRC are recall and precision, respectively. Average precision (AP) is calculated by using the integral area of the PRC. AP was the average of the precision obtained by IoU at intervals of 0.05 from 0.5 to 0.95. The AP is calculated as follows:

$$\text{AP} = \int_0^1 P(R) dR. \quad (8)$$

Among them,  $R$  represents the recall rate and  $P$  represents the precision. AP50 is the AP calculated when IoU was 0.5. mAP is the average of multiple categories of AP.

The evaluation metrics of object detection that are usually used are precision, recall, and AP.

The model size and the FLOPs are usually used for evaluating the performance of the running speed of the detector. For a convolution layer, suppose its size is  $h \times w \times c \times n$ , where  $c$  is the number of input channels,  $n$  is the number of output channels, and the size of the output characteristic diagram is  $h \times w$ .

The parameter quantity of the convolution layer is

$$\#\text{params} = n \times (h \times w \times c + 1). \quad (9)$$

The FLOPs of the convolution layer is

$$\#\text{FLOPs} = H \times W \times n \times (h \times w \times c + 1). \quad (10)$$

But FLOP is an indirect indicator. The direct metric is the speed or latency, or known as FPS that we really care about. The correlation between delay and FLOPs and parameter quantities is weak. For example, ShuffleNetv2 has a high number

of parameters but a low latency. The discrepancy between the indirect and direct metric can be attributed to two main reasons. First, MAC constitutes a large portion of runtime in certain operations like group convolution. This cost should not be simply ignored during network architecture designing. Second, some operators are not optimized for the hardware, for example, depthwise convolution, pointwise convolution,  $1 \times 1$  convolution, and so on. They have small model size and few parameters but do not run fast. Therefore, using the indirect metric for computation complexity is insufficient and could lead to suboptimal design. So when designing real-time object detection algorithm, we should not consider indirect indicators, but also consider direct indicators.

#### D. Anchor-Free-Based SAR Ship Detectors

Section IV-C describes the motivation and advantages of anchor-free detectors. It also indicates that the anchor-free detectors are especially suitable for SAR ship detection. This is because the ships in SAR images are very sparse, most anchor boxes are redundant and will lead to the computation burden. What is more, the sizes of ships in SAR images are small, anchor boxes are hard to match them with the ground-truth, which leads to poor performance on small ships in large scenes. Last but not the least, the anchor-free detectors have great potential to realize real-time SAR ship detection with high accuracy. So a lot of researchers use anchor-free ideas to detect ship in SAR images. We will survey them in the following part. There are 18 papers that are anchor-free-based SAR ship detector. They are shown in Table VII.

Table VII shows the dates, authors, titles, journals/conferences, datasets, and performances of the 18 papers. We can find that the earliest paper is published at April 3, 2020, which is far later from the first dataset was open to the public at December, 1 2017. This is because the anchor-free detectors are proposed and get popular at 2019. After the anchor-free detectors appear in large numbers, the researchers of SAR ship detection draw lessons from them gradually. So the date is later than 2019.

We can also find that there are 16 journals and 2 conferences among the 18 papers. The two most frequent journals are *Remote Sensing* and *IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING*. This phenomenon shows that the anchor-free detectors are advanced and highly innovative.

Table VII also shows that SSDD and HRSID are the two most frequently used datasets. The AP50 on SSDD vary from 90.75% to 97.8%, and the AP50 on HRSID vary from 88.39% to 97.8%. The numbers of parameters vary from 32.1M to 0.83M. The test times vary from 64 to 7.01 ms. The model sizes vary from 228 to 10.3 MB. The FLOPS vary from 30.64G to 18.38G.

We further classify the above 18 papers into four classes, as shown in Fig. 25 below. They are CenterNet-based SAR ship detectors, FCOS-based SAR ship detectors, YOLOX-based SAR ship detectors, and others. Next, we summarize them, respectively.

1) *CenterNet Based*: CenterNet estimates the center point of the bounding box and regresses the width and height of the object. It is an anchor free, end-to-end differentiable, simpler,

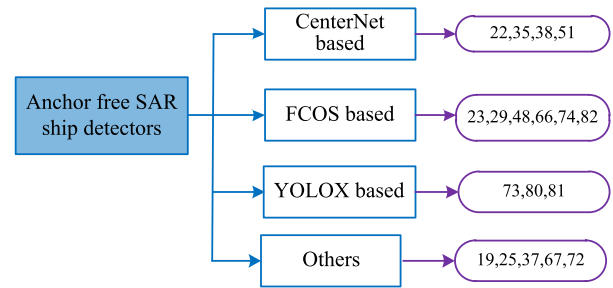


Fig. 25. Anchor free SAR ship detectors.

and accurate detector compare with the anchor-based detectors. It achieves 37.4% AP at 52 FPS on the MS COCO dataset. CenterNet predicts the center point and only local peaks are extracted in the keypoint heatmap. The NMS is abandoned, which can accelerate the speed. It has a higher output resolution (output stride of 4), which are helpful for small targets. Due to the advantages of CenterNet, there are four papers that are based on CenterNet and improve it according to the properties of ships in SAR images.

Cui et al. [22] introduced the spatial shuffle-group enhance (SSE) attention module to improve the performance of CenterNet. SSE divides the feature map into  $G$  groups along the channel. The channels then are shuffled to improve the interaction of different groups. By this, the stronger semantic features are extracted, which can suppress the noise to reduce false positives. The experiments on SAR-Ship dataset demonstrate the effective of the proposed method. It achieved 94.7% AP50 with 18 ms testing time on 1080Ti GPU. Guo et al. [35] improved CenterNet with three modules, and the new detector called CenterNet++. The feature refinement module is used to extract multiscale information, which is helpful for detecting small ships. The feature pyramids fusion module is used for producing features with more semantic information. The head enhancement module is used for balancing the ratio of foreground and background. CenterNet++ achieved AP50 of 73.9%, 95.1%, and 95.4% on AIR-SARShip, SSDD, and SAR-Ship, respectively, with 33 ms testing time on TITAN RTX GPU. Wang et al. [38] introduced the spatial groupwise enhance (SGE) attention module to CenterNet to detect the dense docked SAR ships. SGE uses the similarity of global and local features to generate attention mask. It has strong semantic information. It reduces the calculations and improves the spatial features of each group. It achieved AP50 of 93.9% on SAR-Ship with GTX 1080Ti GPU. Jiang et al. [51] proposed R-Centernet+ which dedicated to solve the problem of sparse and small properties of ships in SAR images with rotatable bounding box. The convolutional block attention module is used to improve attention of the small ships. The foreground enhance module is used to reject the disturbing of the background. It achieved AP50 of 95.11% and 84.89% on SSDD and AIR-SARShip with 30.64GFLOPS, 16.31M parameters, 77.92 model size, and 33 FPS on NVIDIA 2070 Super GPU.

2) *FCOS Based*: FCOS is a fully convolutional single-stage anchor-free detector. It predicts objects in a per-pixel way. The anchor-free idea avoids the computation about anchor boxes

TABLE VII  
EIGHTEEN ANCHOR FREE BASED SAR SHIP DETECTION PAPERS

No.	Date	Author	Title	Journal/Conference	Dataset	Performance
19	Apr 3 2020	Yuxing Mao et al.	Efficient Low-Cost Ship Detection for SAR Imagery Based on Simplified U-Net	IEEE Access	SSDD	AP: 68.1% AP50: 94% Params: 0.93 M
22	May 20 2020	Zongyong Cui et al.	Ship Detection in Large-Scale SAR Images Via Spatial Shuffle-Group Enhance Attention	IEEE Transactions on Geoscience and Remote Sensing	SAR-Ship-Dataset	AP: 94.7% Testing Time: 18 ms
23	Jul 7 2020	Jiamei Fu et al.	An Anchor-Free Method Based on Feature Balancing and Refinement Network for Multiscale Ship Detection in SAR Images	IEEE Transactions on Geoscience and Remote Sensing	SSDD AIR-SARShip-1.0	AP50: 94.1% AP50: 84.6% Params: 32.5 M Time: 40.1 ms AP: 86.99% Params: 0.83 M Testing Time: 33 ms
25	Aug 13 2020	Fei Gao et al.	Anchor-free Convolutional Network with Dense Attention Feature Aggregation for Ship Detection in SAR Images	Remote Sensing	AIRSARShip-1.0	AP: 61.5% Params: 1.17 M
29	Oct 14 2020	Yuxing Mao et al.	An Anchor-free SAR Ship Detector with Only 1.17M Parameters	International Conference on Aviation Safety and Information Technology (ICASIT)	SSDD	AP50: 73.9% AP50: 95.1% AP50: 95.4% Testing Time: 33 ms
35	Dec 28 2020	Haoyuan Guo et al.	A CenterNet++ model for ship detection in SAR images	Pattern Recognition	AIR-SARShip-1.0 SSDD SAR-Ship-Dataset	AP50: 90.75%
37	Feb 3 2021	Quanzhi An et al.	Transitive Transfer Learning-Based Anchor Free Rotatable Detector for SAR Target Detection With Few Samples	IEEE Access	SSDD	AP50: 93.9%
38	Feb 17 2021	Xiaoya Wang et al.	Dense Docked Ship Detection via Spatial Group-Wise Enhance Attention in SAR Images	IEEE International Geoscience and Remote Sensing Symposium (IGARSS)	SAR-Ship-Dataset	AP50: 96.01% Testing Time: 64 ms Params: 32.1 M Size: 228 MB
48	Jul 26 2021	Zhongzhen Sun et al.	An Anchor-free Detection Method for Ship Targets in High-Resolution SAR Images	IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing	HRSID	AP50: 95.11% AP50: 84.89% FPS: 33 FLOPs: 30.64 G Params: 16.31 M Size: 77.92 MB AP50: 97.8% FPS: 17.5 AP50: 75.5% FPS: 52
51	Aug 24 2021	Yuhang Jiang et al.	R-CenterNet+: Anchor-Free Detector for Ship Detection in SAR Images	Sensors	SSDD AIR-SARShip-1.0	AP50: 95% FPS: 14
66	Feb 25 2022	Mingming Zhu et al.	A Ship Detection Method via Redesigned FCOS in Large-Scale SAR Images	Remote Sensing	SSDD LS-SSDD-v1.0	AP50: 96.5% AP50: 92% Time: 0.356s@ CPU
67	Feb 25 2022	Qi Hu et al.	BANet: A Balance Attention Network for Anchor-Free Ship Detection in SAR Images	IEEE Transactions on Geoscience and Remote Sensing	HRSID	AP50: 97.4% AP50: 89.7% AP: 64.4% FLOPs: 18.38 G Params: 5.68 M Testing Time: 7.01 ms AP50: 89.8% AP: 68.5% FPS: 60.8
72	Apr 4 2022	Bokun He et al.	An Anchor-Free Method Based on Adaptive Feature Encoding and Gaussian-Guided Sampling Optimization for Ship Detection in SAR Imagery	Remote Sensing	SSDD HRSID	AP50: 91.76% Testing Time: 11 ms AP50: 95.5% AP: 67.47% AP50: 88.39% AP: 63.66% Size: 10.3 MB Testing Time: 7.1 ms AP50: 96.35% AP50: 89.74% Size: 136 MB Testing Time: 32 ms
73	Apr 15 2022	Yun Feng et al.	A Lightweight Position-Enhanced Anchor-Free Algorithm	Remote Sensing	SSDD HRSID	
74	Apr 23 2022	Mingming Zhu et al.	A Novel Anchor-Free Method Based on FCOS + ATSS for Ship Detection in SAR Images	Remote Sensing	HRSID	
80	Jul 4 2022	H. Peng et al.	Improved YOLOX's Anchor-Free SAR Image Ship Target Detection	IEEE ACCESS	HRSID	
81	Jul 10 2022	Wenbo Yu et al.	A Lightweight Network Based on One-Level Feature for Ship Detection in SAR Images	Remote Sensing	SSDD HRSID	
82	June 21, 2022	Man Xiao et al.	Power Transformations and Feature Alignment Guided Network for SAR Ship Detection	IEEE Geoscience and Remote Sensing Letters	SSDD HRSID	

and hyper-parameters fine-tuning. It also reduces the computation. FCOS introduces pixel-by-pixel regression prediction, multiscale features, and centerness to improve the performance. It achieved 41.0% AP and 60.7% with ResNet-101-FPN backbone on MS COCO.

Fu et al. [23] proposed feature balancing and refinement network (FBR-Net) to solve the sparsity and anchor settings problem of anchor base detector. FBR-Net improved FCOS in the following aspects. It uses attention-guided balanced pyramid to fuse features along different levels, which are helpful for detecting small ships in SAR images. The feature refinement

module is used to prevent the interference near the ship, which can improve the accuracy of localization. FBR-Net achieved 94.1% and 84.6% AP50 on SSDD and AIR-SARShip-1.0, respectively, with 32.5M parameters and 40.1 testing time on RTX 2080Ti GPU. Mao et al. [29] proposed ResSARNet with 0.69M parameters as the backbone. And it improved FCOS in four aspects. They are centerness on bounding box regression branch, center sampling, generalized intersection over union, and adaptive training sample selection. It achieved 61.5% AP with 1.17M parameters on SSDD and GTX 1080 GPU. Sun et al. [48] improved FCOS with category-position (CP) module



to make it more suitable for detecting small and complex ships. CP can produce guidance vector from classification branch to improve the localization performance. The classification and regression are redesigned to prevent the interference of fuzzy areas. It achieved 96.01% AP50 with 32.1M parameters, 228 MB model size, 64 ms testing time on HRSID, and GTX 1080Ti GPU. Zhu et al. [66] used FCOS to improve the ability of detecting sparsity, small, and interference ships. A new sample definition is proposed to replace the IoU according to the characteristics of ships. It is effective for improving the accuracy. The same resolution feature convolution module, multiresolution feature fusion module (FF-Module), and feature pyramid module are proposed to improve the feature representation for small ships. The complete intersection over union loss is used to improve the localization accuracy. It achieved 97.8% and 75.5% AP50 on SSDD and LS-SSDD-v1.0, respectively, with 32 FPS on RTX 2080Ti GPU. Zhu et al. [74] used FCOS and ATSS to improve the ability of detecting small ships and ships under complex scattering interferences. The improved residual module and deformable convolution are used in backbone to improve the performance of feature extracting. The combined classification score and localization quality is used to address the inconsistent problem. It achieved 89.8% AP50 on HRSID with 60.8FPS on RTX 2080Ti GPU. Xiao et al. [82] proposed power transformations and feature alignment guided network to extract multiscale features. The power-based convolution block is used for suppressing speckle noise. The feature alignment block is used for avoid the dislocation problems. Experiments on SSDD and HRSID show that it can achieve 96.35% AP50, 89.74% AP50 with 136 MB model size, and 31 FPS on NVIDIA RTX 2080Ti GPUs.

3) *YOLOX Based*: YOLOX is an anchor-free detector with decoupled head and the leading label assignment strategy SimOTA. It achieves good results on accuracy and speed. For example, YOLOX-L has the same number of parameters with YOLOv5-L, but it achieves 50.0% AP on COCO with 68.9 FPS on Tesla V100, surpassing YOLOv5-L by 1.8% AP.

Feng et al. [73] proposed a lightweight position-enhanced anchor-free SAR ship detection algorithm called LPEDet based on YOLOX. The lightweight backbone called NLCNet with separable convolution is used for balancing the speed and accuracy. The position-enhanced attention strategy is used for suppressing clutter by adding position information to the channel attention. It achieved 97.4% and 89.7% AP50 on SSDD and HRSID, respectively, with 18.38G FLOPs, 5.68M parameters, and 7.01 ms testing time on RTX 2060 GPU. Peng et al. [80] proposed an anchor-free detector for detecting small sparse and dense arranged ships. It used ICEIoU to improve the regression. The adaptive-NMS and atrous convolution are used to improve the performance further. It achieved 91.76% AP50 and 11 ms testing time on HRSID and RTX3060ti GPU. Yu et al. [81] proposed a lightweight ship detector based on YOLOX. It only uses one-level of the FPN to get a higher efficiency. The receptive field and the semantic information of the one-level feature are expanded to relieve the decrease of accuracy. By four branches with different dilation rates, it can capture various ships in complex backgrounds. The center-based uniform matching is used

to tackle the imbalance problem in training stage. It achieved 95.5% and 88.39% AP50 on SSDD and HRSID, respectively, with 10.3 MB model size and 7.1 ms testing time on Quadro P6000 GPU.

4) *Others*: Mao et al. [19] proposed an anchor-free detector to improve the efficiency and avoid the massive hyper-parameters. Its backbone is based on the simplified U-Net. It only contains 0.47 million learnable weights. It achieved 94% AP50 on SSDD with 0.93M parameters. In order to avoid tuning of anchor-related parameters, reduce the computation, and improve the results of small ships. Gao et al. [25] proposed an anchor-free detector with dense attention feature aggregation. The inverted residual blocks with depthwise separable convolution, dense attention feature aggregation, spatial and channel squeeze, and excitation block are proposed to improve the feature extracting ability of the detector. It achieved 86.99% AP50 on AirSARShip-1.0 with 0.83M parameters and 33 ms testing time on Tesla K20c GPU. An et al. [37] proposed an anchor-free rotatable detector with flexible structure for ships in SAR images. It achieved 90.75% AP50 on SSDD. Hu et al. [67] proposed an anchor-free balance attention network to improve the accuracy and generalization ability for multiscale ship detection. The local attention module is used based on the deformable convolution to obtain local information of ships. The nonlocal attention module is used to extract the nonlocal features of the SAR image. It achieved 95% AP50 on HRSID with 14 FPS. He et al. [72] proposed an anchor-free detector to detect small ships. The adaptive feature encoding module uses deep semantic features into shallow layers and realizes the adaptive learning of the spatial fusion weights. The Gaussian guided detection head is used to assigning different weights to the detected bounding boxes at different locations in the training process. It achieved 96.5% and 92% AP50 on SSDD and HRSID, respectively, with 0.356 s testing time on CPU.

#### E. Trained From Scratch Based SAR Ship Detector

In order to solve the problems of transfer learning, detectors trained from scratch are proposed, for example, DSOD, DetNet, ScratchDet, and so on. The model trained from scratch not only has high accuracy, but also greatly reduces the size and amount of calculation of the model. Due to these advantages, it is also used in SAR ship detection. There are 11 papers those are trained from scratch. They are shown in Table VIII.

Table VIII shows the dates, authors, titles, journals/conferences, datasets, and performances of the 11 papers. We can find that the earliest paper is published at February 4, 2019. And the first trained from scratch detector in computer (DSOD) is proposed at 2018. We can also find that all the 11 papers are journals. This phenomenon shows that the trained from scratch detectors are advanced and highly innovative.

Table VIII also shows that SSDD are the most frequently used dataset. The AP50 on SSDD vary from 80.12% to 97.2%. The numbers of parameters vary from 18.4M to 0.7M. The test times vary from 41.5 to 4.51 ms. The model sizes vary from 72.1 to 0.69 MB. The FLOPs vary from 30.64G to 18.38G.

TABLE VIII  
ELEVEN TRAINED FROM SCRATCH SAR SHIP DETECTION PAPERS

No.	Date	Author	Title	Journal/Conference	Dataset	Performance
13	Feb 4 2019	Zhipeng Deng et al.	Learning Deep Ship Detector in SAR Images From Scratch	IEEE Transactions on Geoscience and Remote Sensing	OpenSARShip	AP: 73% Params: 18.4 M
16	Sept 23 2019	Xiaohan Zhang et al.	A Lightweight Feature Optimizing Network for Ship Detection in SAR Image	IEEE Access	SSDD	AP50: 80.12% Time: 9.28 ms AP: 97.07% FPS: 233 Params: 165,473 FLOPs: 329,998 Model size : 0.82 MB AP50: 96.08% FPS: 222 FLOPs: 203570 Model Size: 0.69 MB Testing Time: 4.51 ms AP: 81.17% Time: 24 ms Params: 18.9 M MACC: 19.73 G Model Size: 72.1 MB AP: 79.79% Testing Time: 41.5 ms Params: 8.22 M MACC: 7.94 G
21	Apr 22 2020	Tianwen Zhang et al.	ShipDeNet-20: An Only 20 Convolution Layers and <1-MB Lightweight SAR Ship Detector	IEEE Geoscience and Remote Sensing Letters	SSDD	AP50: 73.9% AP50: 95.1% AP50: 95.4% Time: 33 ms AP50: 93.78 FPS: 202 Params: 228246 FLOPs: 456042 Model Size: 1 MB Precision: 92.1% Recall: 87.5% F1: 89.7% FPS: 9.7 AP: 61.5% AP50: 97.2% AP: 60.5% AP50: 90.7% Params: 0.7 M FLOPs: 3.7 G Model size: 5.6 MB AP50: 91.76% Testing Time: 11 ms
24	Jul 21 2020	Tianwen Zhang et al.	HyperLi-Net: A hyper-light deep learning network for high-accurate and high-speed ship detection from synthetic aperture radar imagery	ISPRS Journal of Photogrammetry and Remote Sensing	SSDD	
26	Aug 21 2020-21	Long Han et al.	Asymmetric and Square Convolutional Neural Network for SAR Ship Detection from Scratch	International Conference on Biomedical Signal and Image Processing	SSDD	
31	Oct 30 2020	Long Han et al.	Asymmetric Convolution-Based Neural Network for SAR Ship Detection from Scratch	International Conference on Computing and Pattern Recognition	SSDD+ SAR-Ship-Dataset	
35	Dec 28 2020	Guo H. et al.	A CenterNet++ model for ship detection in SAR images	Pattern Recognition	AIR-SARShip SSDD SAR-Ship-Dataset	
40	Feb 17 2021	Tianwen Zhang et al.	Shipdenet-18: An Only 1 Mb With Only 18 Convolution Layers Light-Weight Deep Learning Network For Sar Ship Detection	IEEE International Geoscience and Remote Sensing Symposium (IGARSS)	SSDD	
43	Apr 27 2021	Kai Zhao et al.	Ship detection from scratch in Synthetic Aperture Radar (SAR) images	International Journal of Remote Sensing	SSDD	
47	Jul 13 2021	Kun Sun et al.	DSDet: A Lightweight Densely Connected Sparsely Activated Detector for Ship Target Detection in High-Resolution SAR Images	Remote Sensing	SSDD HRSID	
80	Jul 4 2022	Hui Peng et al.	Improved YOLOX's Anchor-Free SAR Image Ship Target Detection	IEEE ACCESS	HRSID	

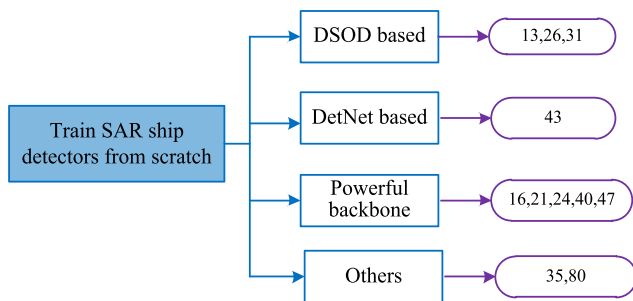


Fig. 26. Detectors trained from scratch in SAR images.

We further classify the above 11 papers into 4 classes, as shown in Fig. 26 below. They are DSOD-based SAR ship detectors, CenterNet-based SAR ship detectors, DetNet-based SAR ship detectors, and others.

DSOD is the first detector that is trained from scratch. It summarized several principles for training detectors from scratch. They are deep supervision, anchor free, stem block, and dense prediction structure. It achieved better results than other detectors with smaller models. Inspired by these ideas, Deng et al. [13] and Han et al. [26] and [31] proposed several methods to improve the performance of the trained from

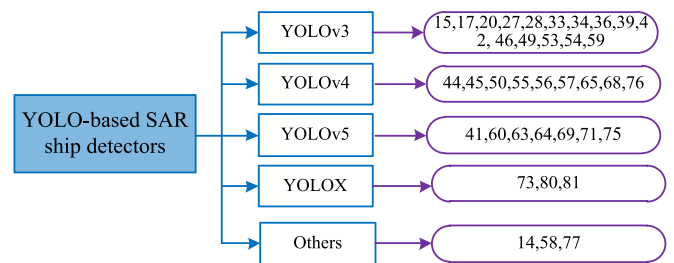


Fig. 27. YOLO-based SAR ship detectors.

detectors on SAR ship detection. Deng et al. [13] proposed the condensed backbone that made the earlier layers receive additional supervision from the objective function, which make easy to train. It can be freely designed and trained from scratch without a large amount of SAR images. The feature reusing strategy, cross-entropy loss, and the position-sensitive score maps are used to improve the performance further. It achieved 73% AP on OpenSARShip with 18.4M parameters. Han et al. [26] proposed an asymmetric and square convolution block to SSD. It can be trained from scratch with less parameters and computations without serious damage to detection accuracy. It achieved 81.17% AP on SSDD with 18.9M parameters, 19.73G

FLOPs, and 72.1 MB model size. Han et al. [31] proposed asymmetric and square convolution feature aggregation block, asymmetric and square convolution feature fusion block to DSOD. It achieved 79.79% AP on SSDD+SAR-Ship-Dataset with 8.22M parameters, 7.94G FLOPs.

DetNet is a backbone network specially designed for object detection. It includes the extra stages against traditional backbone network for image classification, while it maintains high spatial resolution in deeper layers. Due to the advantage of DetNet, Zhao et al. [43] used it and stacked convolution to solve the problem of small object detection. It achieved 92.1% precision, 87.5% recall, and 89.8% F1 score on SSDD with 9.7 FPS.

Zhang et al. [16] designed a lightweight feature optimizing network, which can be trained from scratch and can reduce the testing time without accuracy cost. It used a simpler structure LSSD, a bidirectional FF-Module, and attention mechanism to realize the above purpose. The experiments on SSDD shows that it has 80.12% AP and 9.28 ms testing time on GTX 1080Ti with  $300 \times 300$  input.

Zhang et al. [21], [24], and [40] and Sun et al. [47] designed ShipDeNet-20, ShipDeNet-18, HyperLi-Net, and DSDET, respectively, to detect ships in SAR images by training from scratch. We will review them in Section V-H. Besides, the above papers, Guo et al. [35] and Peng et al. [80] adopted CenterNet and YOLOX as the basic detector to train SAR ship detectors from scratch.

Through the above analysis, we know that the core of training from scratch is to design a good backbone network. Because training from scratch requires the backbone network to have strong feature expression ability and strong supervision information. In SAR ship detection, there are also a lot of papers that train detectors from scratch by designing backbone elaborately. In the next, we will review them.

### F. YOLO Series Based SAR Ship Detector

The two-stage detectors are seldom used in real-time SAR ship detection as the heavy computation. Most of the real-time SAR ship detectors are single-stage. Among them, YOLO series algorithms are naturally designed for real-time detection. Therefore, for real-time SAR ship detection, many researchers use YOLO-based algorithms, which are mainly summarized here. There are 37 papers those are YOLO based SAR ship detector. They are shown in Table IX. Table IX shows the dates, authors, titles, journals/conferences, datasets, and performances of the 36 papers. We can find that most of the papers are among 2020–2021. And SSDD are the most frequently used dataset. The AP50 on SSDD vary from 88.04% to 99.1%. The numbers of parameters vary from 42.6M to 0.857M. The test times vary from 228 to 3.9 ms. The model sizes vary from 31.34 to 2.38 MB.

We further classify the above 37 papers into five classes, as shown in Fig. 27. We can find that YOLOv1 and YOLOv2 are used less, and YOLOv3 is used more. This is because they were proposed in 2015, 2016, and 2018 respectively, and deep

learning was introduced into SAR ship detection in December 2017. After December 2017, researchers will of course adopt more advanced YOLOv3 instead of YOLOv1 and YOLOv2. Second, because YOLOv3 is the most innovative and has a better result, YOLOv4 and YOLOv5 have not made major changes to the network structure. In the following content, we will survey the corresponding papers.

The innovations of YOLOv3 can be summarized as the DarkNet-53+CSP backbone, the FPN neck, and the multibranch prediction. Researchers in this area also improve it in the above components.

Zhang et al. [15] proposed grid CNN based on YOLO and depthwise separable convolution. It has a backbone CNN and a detection CNN. It improved the detection speed. It achieved 90.16% AP50 and 10.94 ms testing time on NVIDIA GTX1080 GPU and SSDD. Zhang et al. [17] proposed depthwise separable convolution neural network for high-speed SAR ship detection. It has a depthwise convolution and a pointwise convolution. The multiscale mechanism, concatenation, and anchor box mechanism are also used. It improved the detection speed. It achieved 94.13% AP50 and 9.03 ms testing time on NVIDIA GTX2080 GPU and SSDD. In order to realize real-time SAR ship detection, Zhang et al. [20] improved YOLOv3 in the following aspects: reduce the size of network, delete the repeated layers, and add two feature concatenation paths. It achieved 90.08% AP50 on SSDD.

Li et al. [27] improved YOLOv3 by adopting dense connection and spatial separation FPNs. It reduces parameters and optimizes the network. Zhou et al. [28] proposed Lira-YOLO based on LiraNet. The backbone LibraNet includes the dense connections, residual connections and group convolution, and stem blocks. The prediction uses a two-layer YOLO prediction layer and adds a residual module for better feature delivery. It achieved 2.980 Bflops, 4.3 MB model size, and 85.46% AP50 on SSDD. Wang et al. [34] proposed SSS-YOLO for detecting multiscale ships. The backbone is redesigned for enriching the spatial and semantics information. The path argumentation fusion network is used to fuse the up and down information. They enhance the detection for small ships. It achieved 67.24% AP and 25.84 ms test time on SAR-Ship-Dataset. Chen et al. [42] used predefined anchor boxes, Darknet-53 with residual units, top-down pyramid structure, soft NMS, mix-up, mosaic, multiscale training, and hybrid optimization to balance the accuracy and speed of SAR ship detector. It achieved 95.52% AP50 with 72 FPS on SSDD and Tesla V100 GPU on SSDD. Hong et al. [46] improved the YOLOv3 in the following aspects. The anchor boxes are redesigned by linear scaling based on the k-means++ algorithm. The Gaussian parameter uncertainty estimators are used for locating. Every scale has four anchor boxes rather three as the difference of ship sizes. It achieved 95.52% AP50 and 21.3 ms test time on SAR-Ship-Dataset. Zhang et al. [53] proposed LSSNet for detecting ships in SAR images. The depthwise separable convolution is used in the early layers, and the stacked dense blocks is used in the deep layers. It achieved high-speed (10.1 test time on GeForce GTX 1660 GPU) and high-accuracy (98.6% AP50 on SSDD). Zhang et al.

TABLE IX  
THIRTY-SEVEN YOLO-BASED SAR SHIP DETECTION PAPERS

No.	Date	Author	Title	Journal/Conference	Dataset	Performance
14	Apr 2 2019	Yang-Lang Chang et al.	Ship Detection Based on YOLOv2 for SAR Imagery	Remote Sensing	SSDD	AP50: 89.76% Testing time: 10.94 ms FLOPs: 44.72 B
15	May 21 2019	Tianwen Zhang et al.	High-Speed Ship Detection in SAR Images Based on a Grid Convolutional Neural Network	Remote Sensing	SSDD	AP50: 90.16% Testing Time: 10.94 ms
17	Oct 24 2019	Tianwen Zhang et al.	Depthwise Separable Convolution Neural Network for High-Speed SAR Ship Detection	Remote Sensing	SSDD	AP50: 94.13% Testing time: 9.03 ms FPS: 111
20	Apr 16 2020	Tianwen Zhang et al.	High-Speed Ship Detection in SAR Images by Improved Yolov3	International Computer Conference on Wavelet Active Media Technology and Information Processing	SSDD	AP50: 90.08%
27	Sept 28 2020	Li Z et al.	Lightweight ship detection methods based on YOLOv3 and DenseNet	Mathematical Problems in Engineering		
28	Oct 6 2020	Long Zhou et al.	Lira-YOLO: a lightweight model for ship detection in radar images	Journal of Systems Engineering and Electronics	SSDD	AP50: 88.04% FLOPs: 2.980 B Params: 4.3 MB
33	Dec 3 2020	Mingming Zhu et al.	Rapid Ship Detection in SAR Images Based on YOLOv3	International Conference on Communication, Image and Signal Processing (CCISP)	SSDD	AP50: 90.69% Testing time: 13.6 ms FPS: 74
34	Dec 14 2020	Jingpu Wang et al.	SSS-YOLO: towards more accurate detection for small ships in SAR image	Remote Sensing Letters	SAR-Ship-Data set	AP: 67.24% Testing Time: 25.84 ms
36	Jan 5 2021	Yash Chaudhary et al.	YOLOv3 Remote Sensing SAR Ship Image Detection	Lecture Notes on Data Engineering and Communications Technologies	SAR-Ship-Data set	
39	Feb 17 2021	Shaobin Jiang et al.	Ship Detection with Sar Based on Yolo	IEEE International Geoscience and Remote Sensing Symposium (IGARSS) 2020	SAR-Ship dataset	F1 score: 93.5%
41	Feb 26 2021	Gang Tang et al.	N-YOLO: A SAR Ship Detection Using Noise-Classifying and Complete-Target Extraction	Remote Sensing	GaoFen-3 dataset	
42	Mar 23 2021	Yao Chen et al.	End-to-End Ship Detection in SAR Images for Complex Scenes Based on Deep CNNs	Journal of Sensors	SSDD	AP50: 95.52% FPS: 72
44	May 13 2021	Jiahuan Jiang et al.	High-Speed Lightweight Ship Detection Algorithm Based on YOLO-V4 for Three-Channels RGB SAR Image	Remote Sensing	SSDD	AP50: 88.08% Model size: 22.5 MB Testing Time: 12.25 ms
45	May 19 2021	Pan Xu et al.	On-Board Real-Time Ship Detection in HISEA-1 SAR Images Based on CFAR and Lightweight Deep Learning	Remote Sensing	SAR-Ship-Data set	AP50: 93.46% FPS: 339 Model size: 22.4 MB
46	Jun 8 2021	Zhonghua Hong et al.	Multi-Scale Ship Detection From SAR and Optical Imagery Via A More Accurate YOLOv3	IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing	SAR-Ship-Data set	AP50: 95.52% Testing time: 21.3 ms
49	Jul 26 2021	Anil Raj J et al.	Lightweight SAR Ship detection and 16 Class Classification using Novel Deep Learning Algorithm with a Hybrid Preprocessing Technique	International Journal of Remote Sensing	SSDD	F1 score: 98.2% Params: 5039316 Model size: 19.21 MB
50	Aug 2 2021	Lin Qiwei et al.	SAR image ship detection based on improved YOLOv4	2021 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA)	SSDD	AP50: 95.2% FPS: 13.94
53	Sept 28 2021	Lianzhong Zhang et al.	Light-SAR-ShipNet (LSSNet):A Lightweight SAR Image Ship Detection System with Enhanced Deep Supervision	ICMTIM 2021	SSDD	AP50: 98.6 Testing Time: 10.1 ms
54	Sept 30 2021	Mingming Zhu et al.	H2Det-A High-Speed and High-Accurate Ship Detector in SAR Images	IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing	SSDD HRSID	AP50: 99.1% Testing Time: 3.6 ms FPS: 278 AP50: 92% Testing time: 3.9 ms FPS: 256
55	Sept 30 2021	Liquan Zhou et al.	A Lightweight YOLOv4 Based SAR Image Ship Detection	2021 IEEE 4th International Conference on Computer and Communication Engineering Technology	SSDD SAR-Ship-Data set	AP50: 95.3% AP50: 95.5%
56	Oct 6 2021	S. Gao et al.	A High-Effective Implementation of Ship Detector for SAR Images	IEEE Transactions on Geoscience and Remote Sensing Letters	HRSID LS-SSDD-V1.0	AP50: 87.49% AP50: 76.2% FLOPs: 104.2 Params: 42.6 M Testing Time: 228 ms 800 × 800
57	Oct 14 2021	Xiaojie Ma et al.	Light-YOLOv4-An Edge-Device Oriented Target Detection Method for Remote Sensing Images	IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing	SSDD	AP50: 93% FPS: 15.12 FLOPs: 5.183 G Model size: 3.5 MB Params: 0.857 M Input size: 416 × 416

TABLE IX  
CONTINUE

58	Oct 19 2021	S. Liu et al.	A Novel Ship Detection Method in Remote Sensing Images via Effective and Efficient PP-YOLO	2021 IEEE International Conference on Sensing, Diagnostics, Prognostics, and Control (SDPC)		
59	Dec 2 2021	Jimin Yu et al.	A Fast and Lightweight Detection Network for Multi-Scale SAR Ship Detection under Complex Backgrounds	Remote Sensing	SSDD SAR-Ship-Data set HRSID	AP50: 97.4% FPS: 42.5 AP50: 96.1% FPS: 60.4 AP50: 88.3% FPS: 24.5
60	Dec 6 2021	Haozhen Zhu et al.	DB-YOLO-A Duplicate Bilateral YOLO Network for Multi-Scale Ship Detection in SAR Images	Sensors	SSDD HRSID	AP50: 97.8% AP: 64.9% AP50: 94.4 % AP: 72.0% Params: 10.8 M FLOPs: 25.6 G FPS: 48.1
63	Feb 6 2022	Kexue Zhou et al.	Ship Detection in SAR Images Based on Multi-Scale Feature Extraction and Adaptive Feature Fusion	Remote Sensing	SSDD SAR-Ship-Data set	AP: 61.1% AP50: 95.6% AP: 60.1% AP50: 95.1% AP: 73.15% FLOPs: 4.44 G Model size: 2.38 MB tested on Jetson TX2
64	Feb 29 2022	Xiaowo Xu et al.	Lite-YOLOv5: A Lightweight Deep Learning Detector for On-Board Ship Detection in Large-Scene Sentinel-1 SAR Images	Remote Sensing	LS-SSDD-v1.0	
65	Feb 24 2022	Tian Miao et al.	Improved YOLOv4 Based on Attention Mechanism for Ship Detection in SAR Images	IEEE ACCESS	SSDD	AP50: 94.16% FPS: 42
68	Feb 25 2022	Shanwei Liu et al.	Multi-Scale Ship Detection Algorithm Based on a Lightweight Neural Network for Spaceborne SAR Images	Remote Sensing	SSDD	AP50: 95.03% FPS: 47.16 Model size: 49.34 MB
69	Mar 3 2022	Xiaowo Xu et al.	On-Board Ship Detection in SAR Images Based on L-YOLO	2022 IEEE Radar Conference	LS-SSDD-v1.0	AP: 73.01% Model size: 7.4 MB FLOPs: 8.1 G
71	Mar 22 2022	M. Xiao et al.	Lightweight SAR Image Target Detection Algorithm Based on YOLO-v5	ISCTT 2021; 6th International Conference on Information Science, Computer Technology and Transportation	SSDD	AP50: 88.7%
73	Apr 15 2022	Yun Feng et al.	A Lightweight Position-Enhanced Anchor-Free Algorithm	Remote Sensing	SSDD HRSID	AP50: 97.4% AP50: 89.7% AP: 64.4% FLOPs: 18.38 G Params: 5.68 M Testing Time: 7.01 ms
75	Apr 28 2022	Fang Xie et al.	Research on the Coordinate Attention Mechanism Fuse in a YOLOv5 Deep Learning Detector for the SAR Ship Detection Task	Sensors	SSDD	AP: 65.6% AP50: 97.0%
76	Apr 30 2022	Jimin Yu et al.	An Efficient Lightweight SAR Ship Target Detection Network with Improved Regression Loss Function and Enhanced Feature Information Expression	Sensors	SSDD HRSID	AP50: 93.56% FPS: 68.52 FLOPs: 3.81 G Model size: 31.34 MB 512 × 512 AP50: 87.06% FPS: 49.67 Model size: 31.34 MB 800 × 800
77	Jun 3 2022	Tian Miao et al.	An Improved Lightweight RetinaNet for Ship Detection in SAR Images	IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing	SSDD	AP50: 98.6% Testing Time: 10.1 ms
80	Jul 4 2022	Hui Peng et al.	Improved YOLOX's Anchor-Free SAR Image Ship Target Detection	IEEE ACCESS	HRSID	AP: 91.76% Time: 11 ms AP50: 95.5% AP: 67.47% AP50: 88.39% AP: 63.66%
81	Jul 10 2022	Wenbo Yu et al.	A Lightweight Network Based on One-Level Feature for Ship Detection in SAR Images	Remote Sensing	SSDD HRSID	Model size: 10.3 MB Testing Time: 7.1 ms

[54] proposed high-speed and high-accurate detector for balancing the accuracy and speed. The fewer convolutional layers, CSP, and rectangle filling is responsible for high speed. The SPP, bottom-up path augmentation, and mosaic data augmentation is responsible for high accuracy. It achieved 95.52% AP, 3.6 ms testing time, 278 FPS on SSDD, and achieved 92% AP50, 3.9 ms testing time, and 256 FPS on HRSID. Yu et al. [59] proposed a fast and lightweight detection network FASC-Net. It includes ASIR-Block, Focus-Block, SPP-Block, and CAPE-Block. It achieved 97.4% AP50 with 42.5 FPS on SSDD, 96.1% AP50 with 60.4 FPS on SAR-Ship-Dataset, and 88.3% AP50 with 24.5 FPS on HRSID. The experiments are conducted on NVIDIA

GTX2070 GPU. Zhu et al. [33] and Yash et al. [36] adopted YOLOv3 without other improvements.

The innovations of YOLOv4 can be summarized as the FPN+PAN+SPP neck and the DIoU Loss. It also uses bag-of-freebies and bag-of-specials to improve the performance further. Researchers in this area also improve it in the above components.

Jiang et al. [44] proposed YOLO-V4-light for real-time SAR ship detection. It greatly reduced the number of convolutional layers in CSPDarkNet53. It only has two prediction branches. YOLO-V4-light decreases from 60 million parameters into 6 million, resulting in a significant reduction in model size and

prediction speed. It achieved 88.08% AP50 22.5MB model size on SSDD. Xu et al. [45] used YOLOv4 as the detector that attached after CFAR obtain more accurate final results. YOLOv4 was not improved at this paper. Lin et al. [50] proposed an improved YOLOv4. The cosine annealing, label smoothing, and mosaic are used. The anchor boxes are selected by K-means clustering on SSDD. It achieved 95.2% AP50 with 13.94 FPS on SSDD on GTX1050TI GPU. It is improved by 2.87% compared with the YOLOv4. Zhou et al. [55] proposed a lightweight YOLOv4 for SAR ship detection. The backbone is the MobileNetv2. The depthwise separable convolution is used for reducing parameters. It achieved 95.5% AP50 on SSDD and the number of parameters is reduced by 40% compare with the original YOLOv4. Gao et al. [56] proposed a high-precision, high-efficiency detector based on YOLOv4. The backbone is SAR-Net, which is similar to CSPDarkNet53 besides the input channel. The neck can balance the relevance of multiscale semantic information for detecting targets of different sizes. The three branches head are redesigned with classification and regression tasks. It achieved 87.49% AP50 on HRSID and 76.2% AP50 on LS-SSDD-V1.0 with 42.6M parameters. Ma et al. [57] compressed the YOLOv4 through sparsity training, pruning, and knowledge distillation. YOLOv4 was not improved at this article. Miao et al. [65] bring attention mechanism into YOLOv4. The threshold attention module is introduced to suppress the adverse effect of complex backgrounds and noises. The channel attention module is embedded into FPN to better enhance the discrimination ability. The decoupled head with two parallel branches improves the performance of classification and regression. It achieved 94.16% AP50 on SSDD with 42 FPS. Liu et al. [68] proposed a lightweight detector based on the YOLOv4-Lite. The backbone is MobileNetv2. The receptive field block is used to improve the feature extraction ability. It achieved 95.03% AP50 with 47.16 FPS and 49.34M model size on SSDD. Yu et al. [76] proposed an efficient lightweight network for SAR ship detection. The ECIoU is proposed to improve the localization accuracy and convergence speed. The SCUPA module is proposed to enhance the multiplexing of picture feature information. The GCHE module is proposed to strengthen the network's ability to extract feature information. It achieved 93.56% AP50 with 68.52 FPS and 31.34M model size on SSDD.

The innovations of YOLOv5 can be summarized as the DarkNet53+Focus+CSP backbone, the FPN+PAN+SPP+CSP neck, and the CIoU Loss.

Tang et al. [41] proposed N-YOLO to detect ships under noises. The YOLOv5 is used after the SAR target potential area extraction module. The YOLOv5 is not improved here. Zhu et al. [60] proposed DB-YOLO to detect small ships and improve the speed. The backbone of DB-YOLO is a single-stage network and has cross-stage partial block, which are helpful for real-time detection. The neck of DB-YOLO used the duplicate bilateral FPN to fuse the semantic and spatial information. The head of YOLO put the bounding boxes and confidence scores as the inputs. It achieved 97.8% AP50 and 64.9% AP on SSDD, and 94.4% AP50 and 72.0% AP on HRSID with 10.8M parameters, 25.6G FLOPs, and 48.1 FPS on RTX 2060 GPU. Zhou et al. [63]

proposed multiscale ship detection network based on YOLOv5s for detecting small ships in SAR images. The cross-stage partial network is used for fusing feature maps adaptively. The FPN with fusion coefficients module is used for choosing the best features to fuse for small ship detection. It achieved 95.6% AP50 and 61.1% AP on SSDD, and 95.1 AP50 and 60.1% AP on SAR-Ship on 2080Ti GPU. Xu et al. [64] and [69] proposed Lite-YOLOv5 and L-YOLO based on YOLOv5 for lightweight on-board SAR ship detection. They have small model size, less FLOPs, and are running on-board without sacrificing accuracy. Xiao et al. [71] proposed YOLO-v5-Light based on YOLO-v5 for detecting ships on the embedded platform. The backbone used separable convolution to reduce the amount of computation, and uses  $1 \times 1$  convolution to fuse channels. The lightweight attention mechanism is also used here. The parameters, model complexity, and weight file size are reduced to 41.4%, 30.3%, and 43.0% of the original network. It achieved 88.7% AP50 on SSDD. Xie et al. [75] proposed YOLO coordinate attention SAR ship for real-time on-board SAR ship detection. It shows advantage in efficiency and performance. It achieved 65.6% AP and 97.0% AP50 on SSDD.

The innovations of YOLOX can be summarized as the decoupled head, the anchor free, and the SimOTA label assign strategy.

Feng et al. [73] proposed lightweight position-enhanced detector. NLCNet is the lightweight feature enhancement backbone with deeply separable convolution. It balanced the speed and accuracy. The position-enhanced attention strategy is used for suppressing background clutter. It achieved 97.4% AP50 on SSDD, 89.7% AP50 on HRSID with 18.38G FLOPs, 5.68M parameters, and 7.01 ms testing time on GeForce RTX2060 GPU. Peng et al. [80] improved YOLOX with corner efficient intersection over union, adaptive-NMS, atrous convolution, and coordinate attention mechanism for detecting sparse and small ships. It achieved 91.76% AP50 with 11 ms testing time on HRSID RTX3060ti GPU. Yu et al. [81] proposed a detector based on YOLOX-s for detecting ships on the board. The one-level feature is used for higher efficiency. The residual asymmetric dilated convolution is used for enlarging the semantic information. The center-based uniform matching is used as the balanced label assignment strategy. It achieved 95.5% AP50 67.47% AP on SSDD, and 88.39% AP50 63.66% AP on HRSID with 10.3 MB model size 7.1 ms testing time On Quadro P6000 GPU.

Besides the above papers, Chang et al. [14], Liu et al. [58], and Miao et al. [77] adopted YOLOv2, PP-YOLO, and RetinaNet to real-time SAR ship detection.

### G. CFAR+CNN Based SAR Ship Detector

In fact, the ocean area occupies the most of the SAR images, and most areas have no ships. We should discard them before inputting these pure backgrounds into deep-learning-based detectors. This can significantly reduce the amount of computation. Fortunately, the CFAR detector can distinguish the pure backgrounds and ship areas with lower computation. Thus, it is necessary to integrate the traditional CFAR methods with

TABLE X  
FOUR CFAR+CNN BASED SAR SHIP DETECTION PAPERS

No.	Date	Author	Title	Journal/Conference	Dataset	Performance
45	May 19 2021	Pan Xu et al.	On-Board Real-Time Ship Detection in HISEA-1 SAR Images Based on CFAR and Lightweight Deep Learning	Remote Sensing	SAR-Ship-Dataset	AP50: 93.46% FPS: 339 Model size: 22.4 MB
62	Feb 3 2022	Xiunan Li et al.	An Improved CFAR-CNN Algorithm for Ship Target Detection	2021 Photonics & Electromagnetics Research Symposium (PIERS)		
64	Feb 20 2022	Xiaowo Xu et al.	Lite-YOLOv5: A Lightweight Deep Learning Detector for On-Board Ship Detection in Large-Scene Sentinel-1 SAR Images	Remote Sensing	LS-SSDD-v1.0	AP: 73.15% FLOPs: 4.44 G Model size: 2.38 MB tested on the Jetson TX2
78	Jun 3 2022	Souad Chabbi et al.	CFAR-CNN Detector of Ships from SAR Image Using Generalized Gamma Distribution and Real Dataset	2022 7th International Conference on Image and Signal Processing and their Applications (ISPA)		

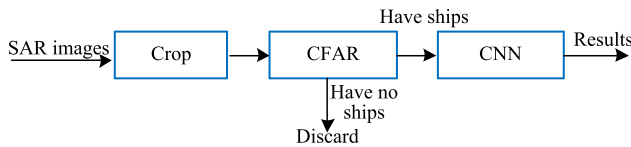


Fig. 28. Process of CFAR- and CNN-based SAR ship detector.

deep-learning-based detector when conducting on-board SAR ship detection. The process of CFAR+CNN is shown in Fig. 28.

Through the above process, the CFAR can quickly exclude chips without ships and prevent CNN with large amount of computation from wasting computing resources. For targeted slices, CNN with high accuracy but slow speed can accurately identify and position them. There are four papers those are based on CFAR+CNN as shown in Table X.

Li et al. [62] proposed CFAR+CNN model for balancing the speed and accuracy of SAR ship detector. The CFAR is used for detecting candidate ship chips, and the CNN is used for removing false alarms generated in the CFAR step. In fact, the combination is the traditional idea. And the deep-learning-based object detectors are not used here. So the accuracy is lower. In order to improve the accuracy and speed of SAR ship detection, Souad Chabbi et al. [78] proposed a CFAR-CNN detector. The generalized gamma distribution is used for modeling the seal clutter. The CNN local detector is applied to improve the accuracy. But it is also traditional combination. Xu et al. [64] proposed Lite-YOLOv5 for detecting ships on-board. It includes a histogram-based pure backgrounds classification module, a shape distance clustering module, a channel and SAM, and a hybrid SPP module to improve detection performance. It is also transplanted into the embedded platform NVIDIA Jetson TX2. Xu et al. [45] combined CFAR and lightweight deep-learning method for detecting ships on-board. CFAR is used for finding potential ships, and YOLOv4 is used for obtain more accurate final results. It achieved 93.46% AP50 on SAR-Ship dataset with 22.4MB model size.

#### H. Lightweight Backbone Networks for SAR Ship Detection

Backbone network is used to extract features, which is an important part of detection algorithm. It occupies a very large amount of computation. Therefore, in order to realize real-time

detection, many researchers are studying how to design a small and powerful backbone network, such as MobileNets, ShuffleNets, and so on. Many similar results have been achieved in real-time SAR ship detection, as shown in Table XI. We will introduce them one by one below.

Zhang et al. [21] designed a lightweight SAR ship detector ShipDeNet-20 with only 0.82 MB model size. The FF-Module, feature enhance module, and scale share feature pyramid module (SSFP-Module) are used to compensate for the accuracy loss. The backbone has 15 layers, FF-Module has 2 layers, and SSFP-Module has 3 layers. All the convolution layers are depthwise convolution, which makes it more lightweight. It is also trained from scratch. It achieved 97.07% AP50, 233 FPS, and 0.82 MB model size on SSDD. Zhang et al. [24] proposed HyperLi-Net for high-accurate and high-speed SAR ship detection. HyperLi-Net used five modules to ensure the accuracy: multireceptive-field, dilated convolution, channel and spatial attention, feature fusion, and feature pyramid. HyperLi-Net used five modules to ensure the speed: region-free, small kernel, narrow channel, separable convolution, and batch normalization fusion. It is also trained from scratch. It achieved 96.08% AP50, 222 FPS, 0.69 model size, and 4.51 testing time on SSDD with RTX2080Ti GPU. Zhang et al. [40] proposed Shipdenet-18 with only 1 Mb model size for lightweight SAR ship detection. It has fewer layers and fewer kernels. The deep and shallow FF-Module and the feature pyramid module are used for fusing different features. It is also trained from scratch. It has 228 246 parameters, 456 042 FLOPs, and 1 MB model size. It achieved 93.78% AP50 on SSDD. Zhang et al. [17] proposed a high-speed SAR ship detector based on depthwise separable convolution. The conventional convolution is substituted by depthwise convolution and pointwise convolution. The multiscale, concatenation, and anchor box mechanism are used for the real-time detection. It achieved 94.13% AP50 with 9.03 ms testing time and 111 FPS on SSDD and NVIDIA RTX2080Ti GPU. Feng et al. [73] proposed LPEDet for real-time SAR ship detection. The backbone discarded the squeeze-and-excitation module and designed a lightweight convolution block. It shows advantages on accuracy and speed than other methods on SSDD and HRSID.

Most of the above papers are based on the depthwise convolution, which is not optimized on the hardware, e.g., GPU, FPGA, and DSP. So although the detection algorithm is lightweight, it does not mean that it can be processed in real time on hardware.

TABLE XI  
FIVE LIGHTWEIGHT BACKBONE NETWORK BASED SAR SHIP DETECTION PAPERS

No.	Date	Author	Title	Journal/Conference	Dataset	Performance
17	Oct 24 2019	Tianwen Zhang et al.	Depthwise Separable Convolution Neural Network for High-Speed SAR Ship Detection	Remote Sensing	SSDD	AP50: 94.13% Testing time: 9.03 ms FPS: 111
21	Apr 22 2020	Tianwen Zhang et al.	ShipDeNet-20: An Only 20 Convolution Layers and <1-MB Lightweight SAR Ship Detector	IEEE Geoscience and Remote Sensing Letters	SSDD	AP50: 97.07% FPS: 233 Params: 165,473 FLOPs: 329,998 Model size: 0.82MB
24	Jul 21 2020	Tianwen Zhang et al.	HyperLi-Net: A hyper-light deep learning network for high-accurate and high-speed ship detection from synthetic aperture radar imagery	ISPRS Journal of Photogrammetry and Remote Sensing	SSDD	AP50: 96.08% FPS: 222 FLOPs: 203570 Model Size: 0.69 MB Testing Time: 4.51 ms
40	Feb 17 2021	Tianwen Zhang et al.	Shipdenet-18: An Only 1 Mb With Only 18 Convolution Layers Light-Weight Deep Learning Network For Sar Ship Detection	EEE International Geoscience and Remote Sensing Symposium (IGARSS)	SSDD	AP50: 93.78% FPS: 202 Params: 228246 FLOPs: 456042 Model Size: 1 MB
73	Apr 15 2022	Yun Feng et al.	A Lightweight Position-Enhanced Anchor-Free Algorithm	Remote Sensing	SSDD HRSID	AP50: 97.4% AP: 89.7% AP: 64.4% FLOPs: 18.38 G Params: 5.68 M Testing Time: 7.01 ms

TABLE XII  
SIX PRUNING, QUANTIZATION, AND KNOWLEDGE DISTILLATION BASED SAR SHIP DETECTION PAPERS

No.	Date	Author	Title	Journal/Conference	Dataset	Performance
18	Jan 7 2020	Shiqi Chen et al.	Learning Slimming SAR Ship Object Detector Through Network Pruning and Knowledge Distillation	IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing	SSDD	AP50: 94.6% FLOPs: 1.94 G FPS: 258.6 Testing Time: 3.9 ms Params: 0.6 M Model size: 2.8 MB
30	Oct 14 2020	Yuxing Mao et al.	Network Slimming Method for SAR Ship Detection Based on Knowledge Distillation	International Conference on Aviation Safety and Information Technology (ICASIT)	SSDD	AP50: 92.6% AP: 56.5% Params: 61.5 M Testing Time: 17.27 ms
57	Oct 14 2021	Xiaojie Ma et al.	Light-YOLOv4-An Edge-Device Oriented Target Detection Method for Remote Sensing Images	IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing	SSDD	AP50: 93% FPS: 15.12 FLOPs: 5.183 G Model size: 3.5 MB Params: 0.857 M Input size: 416 × 416
61	Feb 15 2022	Chen Shiqi et al.	A lightweight, arbitrary-oriented SAR ship detector via feature map-based knowledge distillation	Journal of Radars	HRSID	AP50: 80.71% Params: 9.07 M FPS: 28.76
64	Feb 20 2022	Xiaowo Xu et al.	Lite-YOLOv5: A Lightweight Deep Learning Detector for On-Board Ship Detection in Large-Scene Sentinel-1 SAR Images	remote sensing	LS-SSDD-v1.0	AP: 73.15% FLOPs: 4.44 G Model size: 2.38 MB tested on the Jetson TX2
79	Jun 24 2022	Xi Yang et al.	An Efficient and Lightweight CNN Model With Soft Quantification for Ship Detection in SAR Images	IEEE IEEE Transactions on Geoscience and Remote Sensing	SSDD AIR-SARShip-2.0 SAR-Ship-Dataset	AP50: 97.0% AP50: 77.0% AP50: 95.7% Params: 3.22 M FLOPs: 1.4 G Testing Time: 9.2 ms

Besides the above innovations, researchers should seek other ideas to realize real-time SAR ship detection.

We can also find that training from scratch and lightweight detection network designing are strongly correlated. Training from scratch needs to design the backbone network, and the designed lightweight network generally needs to be trained from scratch to avoid the disadvantages brought by pretraining on ImageNet.

### I. Pruning, Quantization, and Knowledge Distillation on SAR Ship Detection

The deployment of deep CNNs in real-time SAR ship detection is largely hindered by huge storage and computational cost. Model compression and acceleration are necessary approaches to realize real-time target detection. The real-time detection

of ship targets in SAR images also requires pruning quantization and knowledge distillation on large models to achieve lightweight and high accuracy detection models. There are six papers that are about this direction, which are shown in Table XII

Chen et al. [18] slimmed SAR ship detector by pruning and knowledge distillation. The pruning makes the backbone channel-level sparsity. The network weights and scaling factors are jointly trained with L1 regularization in channelwise scheme. The FIR-KD is proposed to make up for the accuracy decline of pruning. It redefines the extracted knowledge as the relationship between different levels of feature maps, and then transfers it from a large network to a smaller network. It achieved 94.6% AP50 1.94G FLOPs 258.6 FPS 3.9 ms testing time 0.6M parameters and 2.8 MB model size on SSDD. Mao et al. [30] also slimmed SAR ship detector by pruning and knowledge distillation. The detector is pruned on filter level to get lightweight models. The Kullback–Leibler divergence based knowledge distillation



TABLE XIII  
SIX PAPERS ABOUT HARDWARE DEPLOYMENT OF SAR SHIP DETECTION PAPERS

No.	Date	Author	Title	Journal/Conference	Dataset	Performance
45	May 9 2021	Pan Xu et al.	On-Board Real-Time Ship Detection in HISEA-1 SAR Images Based on CFAR and Lightweight Deep Learning	Remote Sensing	SAR-Ship-Dataset	AP50: 93.46% FPS: 339 Model size: 22.4 MB
52	Sept 3 2021	Jerzy Stefanowicz	Spaceborne SAR ship-detection system using FPGA SoCs with integrated ADCs/DACs	Photonics Applications in Astronomy, Communications, Industry, and High Energy Physics Experiments 2021		
57	Oct 14 2021	Xiaojie Ma et al.	Light-YOLOv4-An Edge-Device Oriented Target Detection Method for Remote Sensing Images	IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing	SSDD	AP50: 93% FPS: 15.12 FLOPs: 5.183 G Model size: 3.5 MB Params: 0.857 M Input size: 416 × 416
64	Feb 20 2022	Xiaowo Xu et al.	Lite-YOLOv5: A Lightweight Deep Learning Detector for On-Board Ship Detection in Large-Scene Sentinel-1 SAR Images	Remote Sensing	LS-SSDD-v1.0	AP: 73.15% FLOPs: 4.44 G Model size: 2.38 MB tested on Jetson TX2
69	Mar 3 2022	Xiaowo Xu et al.	On-Board Ship Detection in SAR Images Based on L-YOLO	2022 IEEE Radar Conference	LS-SSDD-v1.0	AP: 73.01% Model size: 7.4 MB FLOPs: 8.1 G
70	Mar 22 2022	Geng Yang et al.	Algorithm/Hardware Codesign for Real-Time On-Satellite CNN-Based Ship Detection in SAR Imagery	IEEE Transactions on Geoscience and Remote Sensing	SSDD	

is proposed to train small student network and large teacher network (YOLOv3@EfficientNet-B7) to make up for the accuracy decline. It achieved 92.6% AP50, 56.5% AP, 61.5M parameters, and 17.27 ms testing time on SSDD with GTX 1080Ti GPU. Ma et al. [57] compressed YOLOv4 for designing an edge-device-oriented lightweight detector. The sparsity training on channels and layers is used by L1 regularization. The channel pruning and layer pruning are used to prune the less important parts, which reduce the width and depth. Then knowledge distillation is used to improve the accuracy, and the model is quantized to FP16 to further accelerate the model. At last, it is deployed on NVIDIA Jetson TX2. It achieved 93% AP50, 15.12 FPS, 5.183G FLOPs, 3.5MB model size, and 0.857M parameters on SSDD with 416 × 416 input size on SSDD. Chen et al. [61] proposed a lightweight detector by feature-map-based knowledge distillation. When training the lightweight student network, the similarity between pixels is treated as transferred knowledge in heatmap distillation. It achieved 80.71% AP50 with only 9.07M model parameters on HRSID. Yang et al. [79] proposed an efficient and lightweight detector with soft quantification for real-time SAR ship detection. The split bidirectional FPN is used to compensate for the lack of accuracy. The soft quantization simulates the quantization process of training and learns variable parameters to adjust the pixel value results of each channel, so as to adjust the distribution of the feature map to make it as similar as possible to the original feature map. It achieved 97.0% AP50 on SAR ship detection dataset with less than 15× parameters and less than 6× the FLOPs. Xu et al. [64] designed a lightweight cross-stage partial module to reduce the amount of calculation and pruned it for a more compact detector. The detector is sparse regularization trained first. Then it is pruned to get sparse channels. Finally, the model is fine-tuned to restore the accuracy by iterating the pruning procedures.

### J. Hardware Deployment of SAR Ship Detector

Hardware deployment is the last step of SAR ship detection. GPU is the common hardware for desktop real-time target detec-

tion. NVIDIA TX2 and FPGA are the commonly used hardware for end-to-end real-time object detection. There are six papers that are about hardware deployment of SAR ship detection as shown in Table XIII. We will survey them in the following parts.

Xu et al. [45] proposed an on-board ship detection method based on the CFAR and lightweight deep learning. It can be used by the SAR satellite on-board computing platform. The Jetson TX1 is used as the hardware to realize on-board ship detection. The intelligent application module is used on HISEA-1 satellite. The on-board ship detection method extracts the ship chips and position information, and transmitted to the ground. The model parameters in the satellite's intelligent processing unit can be updated. It shows good results on HISEA-1 SAR images. Jerzy et al. [52] described the modern FPGA SoCs, SAR systems, and on-board detection system. It proposed an SAR ship detection system on SoC-based radar payloads. The paper is concluded with a few observations on how implementing such a system could affect existing radar platforms. Xu et al. [64] proposed Lite-YOLOv5 based on YOLOv5 and other lightweight ideas. It conducted on-board ship detection with 4.44G FLOPs and 73.15% AP50 on S-SSDD-v1.0. It is transplanted to NVIDIA Jetson TX2. Xu et al. [69] proposed lightweight SAR ship detector named L-YOLO. It is also transplanted to NVIDIA Jetson TX2 to validate the practicability. It achieved 73% AP50 with 8.1G FLOPs on LS-SSDD-v1.0. Ma et al. [57] proposed Light-YOLOv4 for edge-device-oriented SAR ship detection. The sparsity training, channel and layer pruning, knowledge distillation, and quantization are used to compress the detector further. It is also deployed on NVIDIA Jetson TX2. The experiments on SSDD show that the detection speed is increased to 4.2×. Yang et al. [70] proposed an algorithm and hardware codesigning method for real-time SAR ship detection. The proposed OSCAR-RT is the first end-to-end algorithm and hardware codesigning method for real-time on-satellite CNN-based SAR ship detection. It proposed a fully pipelined inter layer flow acceleration architecture, in which all layers of CNN model can use FPGA resources on chip for concurrent processing. It proposes a hardware-guided, progressive, and structured pruning method,

which is guided by hardware metrics. The coarse-grained and fine-grained filter pruning and mixed precision quantization are also used. A highly optimized CNN component library is designed. The trimmed CNN model is mapped to these hardware library components in a fully pipelined inter layer flow manner. The proposed method achieves an AP50 of 94% on SSDD, with speed of 652 FPS on Xilinx VC709 FPGA while consuming only 5.8 W power.

## VI. DISCUSSION

Here, we reviewed the papers about real-time SAR ship detection. The model compression and acceleration methods, the real-time object detection methods are introduced first. The years, datasets, journals, deep-learning frameworks, public datasets, and the evaluation metrics of the 70 papers are introduced second. These 70 papers are reviewed from the following aspects: anchor free, trained from scratch, YOLO series, CFAR+CNN, lightweight backbone, model compressing, and hardware deployment. The speed and accuracy show the rapid development of these algorithms in recent years in this field. Based on the above review, we find that the real-time SAR ship detector should have the following attributes: single-stage, trained from scratch, anchor free, lightweight backbone, and head networks, using model compression and acceleration, optimized and transplanted on the edge hardware. Researchers should follow most of the above attributes when realizing real-time SAR ship detection. What is more, due to the fact that ships in SAR images are extremely sparse, and most areas are without targets, the CFAR+CNN show great potential for handle this situation. What is more, the lightweight networks with much depthwise and pointwise convolution will not have a fast speed. As these operations are not optimized on the hardware, e.g., GPU, FPGA, and DSP. So they should be used less. And researchers should both consider the direct (speed) and indirect (FLOPs) indicators. Compared with computer vision, real-time SAR ship detection is less popular, with fewer researchers and less achievements. Most of the achievements are sporadic references from computer vision, with the problem of incomplete innovation. In the future, we should pay more attention and produce more results on this area.

## VII. CONCLUSION

This article gives a comprehensive overview of real-time SAR ship detection. First, we introduce the model compression and acceleration methods. They are the sources of innovation in real-time SAR ship detection. The principle and research status of pruning, quantization, knowledge distillation, low-rank factorization, lightweight networks, and model deployment are introduced in detail. Second, we introduce the real-time object detection methods. The also provide inspirations for real-time SAR ship detection. The two-stage real-time detectors, single-stage real-time detectors, anchor-free detectors, trained from scratch detectors, compression, and acceleration in object detection are introduced, respectively. Third, 70 public papers about deep-learning-based real-time SAR ship detection are reviewed comprehensively. Ten public datasets and the usually

used evaluation metrics are introduced in the beginning. Then 70 papers are categorized into 7 types and are reviewed in detail. They are anchor free based, trained from scratch based, YOLO series based, CFAR+CNN based, lightweight backbone networks based, model compression based, and hardware deployment. The principle, innovation, and performance are reviewed, respectively. Finally, the problems existing in this field and the future direction are described. In the future, we should pay more attention to the lightweight CNN designing, model compression and acceleration, and hardware deployment in this field.

As far as we know, this is the first review on real-time SAR ship detection. It can provide a reference for researchers in this area or who are interested in it. It can make researchers quickly understand the research status.

## ACKNOWLEDGMENT

The authors would like to thank the editors and anonymous reviewers for their valuable comments that can greatly improve this article.

## REFERENCES

- [1] A. Reigber et al., "Very-high-resolution airborne synthetic aperture radar imaging: Signal processing and applications," *Proc. IEEE*, vol. 101, no. 3, pp. 759–783, Mar. 2013, doi: [10.1109/JPROC.2012.2220511](https://doi.org/10.1109/JPROC.2012.2220511).
- [2] P. Iervolino and R. Guida, "A novel ship detector based on the generalized-likelihood ratio test for SAR imagery," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 10, no. 8, pp. 3616–3630, Aug. 2017, doi: [10.1109/JSTARS.2017.2692820](https://doi.org/10.1109/JSTARS.2017.2692820).
- [3] U. Kanjir, H. Greidanus, and K. Ostir, "Vessel detection and classification from spaceborne optical images: A literature survey," *Remote Sens. Environ.*, vol. 207, pp. 1–26, 2018, doi: [10.1016/j.rse.2017.12.033](https://doi.org/10.1016/j.rse.2017.12.033).
- [4] H.-C. Li, W. Hong, Y.-R. Wu, and P.-Z. Fan, "An efficient and flexible statistical model based on generalized Gamma distribution for amplitude SAR images," *IEEE Trans. Geosci. Remote Sens.*, vol. 48, no. 6, pp. 2711–2722, Jun. 2010, doi: [10.1109/TGRS.2010.2041239](https://doi.org/10.1109/TGRS.2010.2041239).
- [5] O. Russakovsky et al., "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, 2015, doi: [10.1007/s11263-015-0816-y](https://doi.org/10.1007/s11263-015-0816-y).
- [6] X. Wu, D. Hong, J. Tian, J. Chanussot, W. Li, and R. Tao, "OR-SIm detector: A novel object detection framework in optical remote sensing imagery using spatial-frequency channel features," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 7, pp. 5146–5158, Jul. 2019, doi: [10.1109/TGRS.2019.2897139](https://doi.org/10.1109/TGRS.2019.2897139).
- [7] X. Wu, D. Hong, J. Chanussot, Y. Xu, R. Tao, and Y. Wang, "Fourier-based rotation-invariant feature boosting: An efficient framework for geospatial object detection," *IEEE Geosci. Remote Sens. Lett.*, vol. 17, no. 2, pp. 302–306, Feb. 2020, doi: [10.1109/LGRS.2019.2919755](https://doi.org/10.1109/LGRS.2019.2919755).
- [8] X. Wu, D. Hong, and J. Chanussot, "UIU-Net: U-Net in U-Net for infrared small object detection," *IEEE Trans. Image Process.*, vol. 32, pp. 364–376, Dec. 2022.
- [9] 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.*, 2017, pp. 1–6, doi: [10.1109/BIGSARDATA.2017.8124934](https://doi.org/10.1109/BIGSARDATA.2017.8124934).
- [10] J. Li, C. Xu, H. Su, L. Gao, and T. Wang, "Deep learning for SAR ship detection: Past, present and future," *Remote Sens.*, vol. 14, 2022, Art. no. 2712, doi: [10.3390/rs14112712](https://doi.org/10.3390/rs14112712).
- [11] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, Art. no. 25.
- [12] V. Sze, Y.-H. Chen, T.-J. Yang, and J. S. Emer, "Efficient processing of deep neural networks: A tutorial and survey," *Proc. IEEE*, vol. 105, no. 12, pp. 2295–2329, Dec. 2017.
- [13] 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, doi: [10.1109/TGRS.2018.2889353](https://doi.org/10.1109/TGRS.2018.2889353).

- [14] Y. L. Chang et al., "Ship detection based on YOLOv2 for SAR imagery," *Remote Sens.*, vol. 11, no. 7, 2019, Art. no. 786, doi: [10.3390/rs11070786](https://doi.org/10.3390/rs11070786).
- [15] T. Zhang and X. Zhang, "High-speed ship detection in SAR images based on a grid convolutional neural network," *Remote Sens.*, vol. 11, no. 10, 2019, Art. no. 1206, doi: [10.3390/rs11101206](https://doi.org/10.3390/rs11101206).
- [16] X. Zhang et al., "Lightweight feature optimizing network for ship detection in SAR image," *IEEE Access*, vol. 7, pp. 141662–141678, 2019, doi: [10.1109/ACCESS.2019.2943241](https://doi.org/10.1109/ACCESS.2019.2943241).
- [17] T. Zhang et al., "Depthwise separable convolution neural network for high-speed SAR ship detection," *Remote Sens.*, vol. 11, no. 21, 2019, Art. no. 2483, doi: [10.3390/rs11212483](https://doi.org/10.3390/rs11212483).
- [18] S. Chen, R. Zhan, W. Wang, and J. Zhang, "Learning slimming SAR ship object detector through network pruning and knowledge distillation," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 14, pp. 1267–1282, Dec. 2021, doi: [10.1109/JSTARS.2020.3041783](https://doi.org/10.1109/JSTARS.2020.3041783).
- [19] 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, doi: [10.1109/ACCESS.2020.2985637](https://doi.org/10.1109/ACCESS.2020.2985637).
- [20] T. Zhang, X. Zhang, J. Shi, and S. Wei, "High-speed ship detection in SAR images by improved Yolov3," in *Proc. 16th Int. Comput. Conf. Wavelet Act. Media Technol. Inf. Process.*, 2019, pp. 149–152, doi: [10.1109/IC-CWAMTIP47768.2019.9067695](https://doi.org/10.1109/IC-CWAMTIP47768.2019.9067695).
- [21] 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, doi: [10.1109/LGRS.2020.2993899](https://doi.org/10.1109/LGRS.2020.2993899).
- [22] Z. Cui, X. Wang, N. Liu, Z. Cao, and J. Yang, "Ship detection in large-scale SAR images via spatial shuffle-group enhance attention," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 1, pp. 379–391, Jan. 2021, doi: [10.1109/TGRS.2020.2997200](https://doi.org/10.1109/TGRS.2020.2997200).
- [23] J. Fu, X. Sun, Z. Wang, and K. Fu, "An anchor-free method based on feature balancing and refinement network for multiscale ship detection in SAR images," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 2, pp. 1331–1344, Feb. 2021, doi: [10.1109/TGRS.2020.3005151](https://doi.org/10.1109/TGRS.2020.3005151).
- [24] T. Zhang et al., "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, no. 12, pp. 123–153, 2020, doi: [10.1016/j.isprsjrs.2020.05.016](https://doi.org/10.1016/j.isprsjrs.2020.05.016).
- [25] H. Zhou, "Anchor-free convolutional network with dense attention feature aggregation for ship detection in SAR images," *Remote Sens.*, vol. 12, 2020, Art. no. 2619, doi: [10.3390/rs12162619/](https://doi.org/10.3390/rs12162619/).
- [26] L. Han et al., "Asymmetric and square convolutional neural network for SAR ship detection from scratch," in *Proc. 5th Int. Conf. Biomed. Signal Image Process.*, 2020, pp. 80–85.
- [27] Z. Li et al., "Lightweight ship detection methods based on YOLOv3 and DenseNet," *Math. Problems Eng.*, vol. 2020, 2020, Art. no. 4813183.
- [28] Z. Long, W. Suyuan, C. Zhongma, F. Jiaqi, Y. Xiaoting, and D. Wei, "Lira-YOLO: A lightweight model for ship detection in radar images," *J. Syst. Eng. Electron.*, vol. 31, no. 5, pp. 950–956, Oct. 2020, doi: [10.23919/JSEE.2020.000063](https://doi.org/10.23919/JSEE.2020.000063).
- [29] Y. Mao et al., "An anchor-free SAR ship detector with only 1.17M parameters," in *Proc. Int. Conf. Aviation Saf. Inf. Technol.*, 2020, pp. 182–186.
- [30] Y. Mao et al., "Network slimming method for SAR ship detection based on knowledge distillation," in *Proc. Int. Conf. Aviation Saf. Inf. Technol.*, 2020, pp. 177–181.
- [31] L. Han et al., "Asymmetric convolution-based neural network for SAR ship detection from scratch," in *Proc. 9th Int. Conf. Comput. Pattern Recognit.*, 2020, pp. 90–95.
- [32] Y. Li, S. Zhang, and W.-Q. Wang, "A lightweight faster R-CNN for ship detection in SAR images," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, Dec. 2022, Art. no. 4006105, doi: [10.1109/LGRS.2020.3038901](https://doi.org/10.1109/LGRS.2020.3038901).
- [33] M. Zhu et al., "Rapid ship detection in SAR images based on YOLOv3," in *Proc. 5th Int. Conf. Commun., Image Signal Process.*, 2020, pp. 214–218, doi: [10.1109/CCISP51026.2020.9273476](https://doi.org/10.1109/CCISP51026.2020.9273476).
- [34] J. Wang et al., "SSS-YOLO: Towards more accurate detection for small ships in SAR image," *Remote Sens. Lett.*, vol. 12, no. 2, pp. 93–102, 2021, doi: [10.1080/2150704X.2020.1837988](https://doi.org/10.1080/2150704X.2020.1837988).
- [35] H. Guo et al., "A CenterNet++ model for ship detection in SAR images," *Pattern Recognit.*, vol. 112, no. 7, 2021, Art. no. 107787, doi: [10.1016/j.patcog.2020.107787](https://doi.org/10.1016/j.patcog.2020.107787).
- [36] Y. Chaudhary et al., "YOLOv3 remote sensing SAR ship image detection," in *Proc. Data Anal. Manage.*, 2021, pp. 519–531.
- [37] Q. An, Z. Pan, H. You, and Y. Hu, "Transitive transfer learning based anchor free rotatable detector for SAR target detection with few samples," *IEEE Access*, vol. 9, pp. 24011–24025, 2021, doi: [10.1109/ACCESS.2021.3056663](https://doi.org/10.1109/ACCESS.2021.3056663).
- [38] X. Wang, Z. Cui, Z. Cao, and S. Dang, "Dense docked ship detection via spatial group-wise enhance attention in SAR images," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2020, pp. 1244–1247, doi: [10.1109/IGARSS39084.2020.9324162](https://doi.org/10.1109/IGARSS39084.2020.9324162).
- [39] S. Jiang, M. Zhu, Y. He, Z. Zheng, F. Zhou, and G. Zhou, "Ship detection with SAR based on Yolo," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2020, pp. 1647–1650, doi: [10.1109/IGARSS39084.2020.9324538](https://doi.org/10.1109/IGARSS39084.2020.9324538).
- [40] T. Zhang, X. Zhang, J. Shi, and S. Wei, "Shipdenet-18: An only 1 Mb with only 18 convolution layers light-weight deep learning network for SAR ship detection," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2020, pp. 1221–1224, doi: [10.1109/IGARSS39084.2020.9323742](https://doi.org/10.1109/IGARSS39084.2020.9323742).
- [41] G. Tang et al., "N-YOLO: A SAR ship detection using noise-classifying and complete-target extraction," *Remote Sens.*, vol. 13, no. 5, 2021, Art. no. 871, doi: [10.3390/rs13050871](https://doi.org/10.3390/rs13050871).
- [42] Y. Chen et al., "End-to-end ship detection in SAR images for complex scenes based on deep CNNs," *J. Sensors*, vol. 2021, 2021, Art. no. 8893182, doi: [10.1155/2021/8893182](https://doi.org/10.1155/2021/8893182).
- [43] K. Zhao, Y. Zhou, X. Chen, B. Wang, and Y. Zhang, "Ship detection from scratch in synthetic aperture radar (SAR) images," *Int. J. Remote Sens.*, vol. 42, no. 13, pp. 5010–5024, 2021, doi: [10.1080/01431161.2021.1906980](https://doi.org/10.1080/01431161.2021.1906980).
- [44] J. Jiang, X. Fu, R. Qin, X. Wang, and Z. Ma, "High-speed lightweight ship detection algorithm based on YOLO-V4 for three-channels RGB SAR image," *Remote Sens.*, vol. 13, 2021, Art. no. 1909, doi: [10.3390/rs13101909](https://doi.org/10.3390/rs13101909).
- [45] P. Xu et al., "On-board real-time ship detection in HISEA-1 SAR images based on CFAR and lightweight deep learning," *Remote Sens.*, vol. 13, 2021, Art. no. 1995, doi: [10.3390/rs13101995](https://doi.org/10.3390/rs13101995).
- [46] Z. Hong et al., "Multi-scale ship detection from SAR and optical imagery via a more accurate YOLOv3," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 14, pp. 6083–6101, Jun. 2021, doi: [10.1109/JSTARS.2021.3087555](https://doi.org/10.1109/JSTARS.2021.3087555).
- [47] K. Sun et al., "DSDet: A lightweight densely connected sparsely activated detector for ship target detection in high-resolution SAR images," *Remote Sens.*, vol. 13, 2021, Art. no. 2743, doi: [10.3390/rs13142743](https://doi.org/10.3390/rs13142743).
- [48] Z. Sun et al., "An anchor-free detection method for ship targets in high-resolution SAR images," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 14, pp. 7799–7816, Jul. 2021, doi: [10.1109/JSTARS.2021.3099483](https://doi.org/10.1109/JSTARS.2021.3099483).
- [49] A. Raj J, S. M. Idicula, and B. Paul, "Lightweight SAR ship detection and 16 class classification using novel deep learning algorithm with a hybrid preprocessing technique," *Int. J. Remote Sens.*, vol. 43, pp. 5820–5847, 2022, doi: [10.1080/01431161.2021.2008544](https://doi.org/10.1080/01431161.2021.2008544).
- [50] L. Qiwei, W. Bowen, and W. Yanfeng, "SAR image ship detection based on improved YOLOv4," in *Proc. IEEE Int. Conf. Artif. Intell. Comput. Appl.*, 2021, pp. 1130–1135, doi: [10.1109/ICAICA52286.2021.9498214](https://doi.org/10.1109/ICAICA52286.2021.9498214).
- [51] Y. Jiang, W. Li, and L. Liu, "R-CenterNet+: Anchor-free detector for ship detection in SAR images," *Sensors*, vol. 21, 2021, Art. no. 5693, doi: [10.3390/s21175693](https://doi.org/10.3390/s21175693).
- [52] J. Stefanowicz, "Spaceborne SAR ship-detection system using FPGA SoCs with integrated ADCs/DACs," in *Proc. SPIE Photon. Appl. Astron., Commun., Ind., High Energy Phys. Experiments*, 2021, Art. no. 120400U, doi: [10.1117/12.2614684](https://doi.org/10.1117/12.2614684).
- [53] L. Zhang et al., "Light-SAR-ShipNet (LSSNet): A lightweight SAR image ship detection system with enhanced deep supervision," *J. Phys.: Conf. Ser.*, vol. 2029, no. 1, 2021, Art. no. 012148.
- [54] M. Zhu, G. Hu, H. Zhou, and S. Wang, "H2Det: A high-speed and high-accurate ship detector in SAR images," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 14, pp. 12455–12466, Nov. 2021, doi: [10.1109/JSTARS.2021.3131162](https://doi.org/10.1109/JSTARS.2021.3131162).
- [55] L.-Q. Zhou and J.-C. Piao, "A lightweight YOLOv4 based SAR image ship detection," in *Proc. IEEE 4th Int. Conf. Comput. Commun. Eng. Technol.*, 2021, pp. 28–31, doi: [10.1109/CCET52649.2021.9544265](https://doi.org/10.1109/CCET52649.2021.9544265).
- [56] S. Gao, J. M. Liu, Y. H. Miao, and Z. J. He, "A high-effective implementation of ship detector for SAR images," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, Oct. 2022, Art. no. 4019005, doi: [10.1109/LGRS.2021.3115121](https://doi.org/10.1109/LGRS.2021.3115121).
- [57] X. Ma, K. Ji, B. Xiong, L. Zhang, S. Feng, and G. Kuang, "Light-YOLOv4: An edge-device oriented target detection method for remote sensing images," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 14, pp. 10808–10820, 2021, doi: [10.1109/JSTARS.2021.3120009](https://doi.org/10.1109/JSTARS.2021.3120009).

- [58] S. Liu, Y. Peng, and L. Liu, "A novel ship detection method in remote sensing images via effective and efficient PP-YOLO," in *Proc. IEEE Int. Conf. Sens., Diagnostics, Prognostics, Control*, 2021, pp. 234–239, doi: [10.1109/SDPCS52933.2021.9563569](https://doi.org/10.1109/SDPCS52933.2021.9563569).
- [59] J. Yu, G. Zhou, S. Zhou, and M. Qin, "A fast and lightweight detection network for multi-scale SAR ship detection under complex backgrounds," *Remote Sens.*, vol. 14, 2022, Art. no. 31, doi: [10.3390/rs14010031](https://doi.org/10.3390/rs14010031).
- [60] H. Zhu, Y. Xie, H. Huang, C. Jing, Y. Rong, and C. Wang, "DB-YOLO: A duplicate bilateral YOLO network for multi-scale ship detection in SAR images," *Sensors*, vol. 21, 2021, Art. no. 8146.
- [61] C. Shiqi et al., "A lightweight, arbitrary-oriented SAR ship detector via feature map-based knowledge distillation," *J. Radars*, vol. 12, no. 1, pp. 140–153, doi: [10.12000/JR21209](https://doi.org/10.12000/JR21209).
- [62] X. Li et al., "An improved CFAR\_CNN algorithm for ship target detection," in *Proc. Photon. Electromagn. Res. Symp.*, 2021, pp. 2057–2062, doi: [10.1109/PIERS53385.2021.9694795](https://doi.org/10.1109/PIERS53385.2021.9694795).
- [63] K. Zhou, M. Zhang, H. Wang, and J. Tan, "Ship detection in SAR images based on multi-scale feature extraction and adaptive feature fusion," *Remote Sens.*, vol. 14, 2022, Art. no. 755.
- [64] X. Xu, X. Zhang, and T. Zhang, "Lite-YOLOv5: A lightweight deep learning detector for on-board ship detection in large-scene sentinel-1 SAR images," *Remote Sens.*, vol. 14, 2022, Art. no. 1018, doi: [10.3390/rs14041018](https://doi.org/10.3390/rs14041018).
- [65] Y. Gao, Z. Wu, M. Ren, and C. Wu, "Improved YOLOv4 based on attention mechanism for ship detection in SAR images," *IEEE Access*, vol. 10, pp. 23785–23797, 2022, doi: [10.1109/ACCESS.2022.3154474](https://doi.org/10.1109/ACCESS.2022.3154474).
- [66] M. Zhu, G. Hu, H. Zhou, S. Wang, Z. Feng, and S. Yue, "A ship detection method via redesigned FCOS in large-scale SAR images," *Remote Sens.*, vol. 14, 2022, Art. no. 1153, doi: [10.3390/rs14051153](https://doi.org/10.3390/rs14051153).
- [67] Q. Hu, S. Hu, and S. Liu, "BANet: A balance attention network for anchor-free ship detection in SAR images," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, Jan. 2022, Art. no. 5222212, doi: [10.1109/TGRS.2022.3146027](https://doi.org/10.1109/TGRS.2022.3146027).
- [68] S. Liu et al., "Multi-scale ship detection algorithm based on a lightweight neural network for spaceborne SAR images," *Remote Sens.*, vol. 14, 2022, Art. no. 1149, doi: [10.3390/rs14051149](https://doi.org/10.3390/rs14051149).
- [69] X. Xu, X. Zhang, T. Zhang, J. Shi, S. Wei, and J. Li, "On-board ship detection in SAR images based on L-YOLO," in *Proc. IEEE Radar Conf.*, 2022, pp. 1–5, doi: [10.1109/RadarConf2248738.2022.9764190](https://doi.org/10.1109/RadarConf2248738.2022.9764190).
- [70] G. Yang et al., "Algorithm/hardware codesign for real-time on-satellite CNN-based ship detection in SAR imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, Mar. 2022, Art. no. 5226018, doi: [10.1109/TGRS.2022.3161499](https://doi.org/10.1109/TGRS.2022.3161499).
- [71] M. Xiao, X. Lv, P. Huang, W. Xu, W. Tan, and Y. Dong, "Lightweight SAR image target detection algorithm based on YOLO-v5," in *Proc. 6th Int. Conf. Inf. Sci., Comput. Technol. Transp.*, 2021, pp. 1–5.
- [72] B. He et al., "An anchor-free method based on adaptive feature encoding and Gaussian-guided sampling optimization for ship detection in SAR imagery," *Remote Sens.*, vol. 14, no. 7, 2022, Art. no. 1738, doi: [10.3390/rs14071738](https://doi.org/10.3390/rs14071738).
- [73] Y. Feng et al., "A lightweight position-enhanced anchor-free algorithm for SAR ship detection," *Remote Sens.*, vol. 14, 2022, Art. no. 1908, doi: [10.3390/rs14081908](https://doi.org/10.3390/rs14081908).
- [74] M. Zhu, G. Hu, S. Li, H. Zhou, S. Wang, and Z. Feng, "A novel anchor-free method based on FCOS + ATSS for ship detection in SAR images," *Remote Sens.*, vol. 14, 2022, Art. no. 2034, doi: [10.3390/rs14092034](https://doi.org/10.3390/rs14092034).
- [75] F. Xie, B. Lin, and Y. Liu, "Research on the coordinate attention mechanism fuse in a YOLOv5 deep learning detector for the SAR ship detection task," *Sensors*, vol. 22, 2022, Art. no. 3370, doi: [10.3390/s22093370](https://doi.org/10.3390/s22093370).
- [76] J. Yu, T. Wu, X. Zhang, and W. Zhang, "An efficient lightweight SAR ship target detection network with improved regression loss function and enhanced feature information expression," *Sensors*, vol. 22, 2022, Art. no. 3447, doi: [10.3390/s22093447](https://doi.org/10.3390/s22093447).
- [77] T. Miao et al., "An improved lightweight RetinaNet for ship detection in SAR images," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 15, pp. 4667–4679, Jun. 2022, doi: [10.1109/JSTARS.2022.3180159](https://doi.org/10.1109/JSTARS.2022.3180159).
- [78] S. Chabbi, F. Farah, and N. Guidoum, "CFAR-CNN detector of ships from SAR image using generalized gamma distribution and real dataset," in *Proc. 7th Int. Conf. Image Signal Process. Appl.*, 2022, pp. 1–6, doi: [10.1109/ISPA54004.2022.9786348](https://doi.org/10.1109/ISPA54004.2022.9786348).
- [79] X. Yang, J. Zhang, C. Chen, and D. Yang, "An efficient and lightweight CNN model with soft quantification for ship detection in SAR images," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, Jun. 2022, Art. no. 5230713, doi: [10.1109/TGRS.2022.3186155](https://doi.org/10.1109/TGRS.2022.3186155).
- [80] H. Peng and X. Tan, "Improved YOLOX's anchor-free SAR image ship target detection," *IEEE Access*, vol. 10, pp. 70001–70015, 2022, doi: [10.1109/ACCESS.2022.3188387](https://doi.org/10.1109/ACCESS.2022.3188387).
- [81] W. Yu, Z. Wang, J. Li, Y. Luo, and Z. Yu, "A lightweight network based on one-level feature for ship detection in SAR images," *Remote Sens.*, vol. 14, 2022, Art. no. 3321, doi: [10.3390/rs14143321](https://doi.org/10.3390/rs14143321).
- [82] M. Xiao, Z. He, X. Li, and A. Lou, "Power transformations and feature alignment guided network for SAR ship detection," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, Jun. 2022, Art. no. 4509405.
- [83] Y. Mao, X. Li, H. Su, Y. Zhou, and J. Li, "Ship detection for SAR imagery based on deep learning: A benchmark," in *Proc. IEEE 9th Joint Int. Inf. Technol. Artif. Intell. Conf.*, 2020, pp. 1934–1940, doi: [10.1109/ITAIC49862.2020.9339055](https://doi.org/10.1109/ITAIC49862.2020.9339055).
- [84] J. Stefanowicz, I. Ali, and S. Andersson, "Current trends in ship detection in single polarization synthetic aperture radar imagery," in *Proc. SPIE, Photon. Appl. Astron., Commun., Ind., High Energy Phys. Experiments*, 2020, Art. no. 1158109, doi: [10.1117/12.2580070](https://doi.org/10.1117/12.2580070).
- [85] T. Zhang et al., "SAR ship detection dataset (SSDD): Official release and comprehensive data analysis," *Remote Sens.*, vol. 13, 2021, Art. no. 3690, doi: [10.3390/rs13183690](https://doi.org/10.3390/rs13183690).
- [86] J. Li, C. Xu, H. Su, L. Gao, and T. Wang, "Deep learning for SAR ship detection: Past, present and future," *Remote Sens.*, vol. 14, 2022, Art. no. 2712, doi: [10.3390/rs14112712](https://doi.org/10.3390/rs14112712).
- [87] L. Jiao et al., "A survey of deep learning-based object detection," *IEEE Access*, vol. 7, pp. 128837–128868, 2019, doi: [10.1109/ACCESS.2019.2939201](https://doi.org/10.1109/ACCESS.2019.2939201).
- [88] L. Liu et al., "Deep learning for generic object detection: A survey," *Int. J. Comput. Vis.*, vol. 128, pp. 261–318, 2020, doi: [10.1007/s11263-019-01247-4](https://doi.org/10.1007/s11263-019-01247-4).
- [89] L. Jiao et al., "A survey of deep learning-based object detection," *IEEE Access*, vol. 7, pp. 128837–128868, 2019, doi: [10.1109/ACCESS.2019.2939201](https://doi.org/10.1109/ACCESS.2019.2939201).
- [90] Y. Cheng et al., "A survey of model compression and acceleration for deep neural networks," 2017, *arXiv:1710.09282*.
- [91] A. Goel, C. Tung, Y.-H. Lu, and G. K. Thiruvathukal, "A survey of methods for low-power deep learning and computer vision," in *Proc. IEEE 6th World Forum Internet Things*, 2020, pp. 1–6, doi: [10.1109/WF-IoT48130.2020.9221198](https://doi.org/10.1109/WF-IoT48130.2020.9221198).
- [92] R. Mishra, H. P. Gupta, and T. Dutta, "A survey on deep neural network compression: Challenges, overview, and solutions," 2020, *arXiv:2010.03954*.
- [93] K. Zhang et al., "Compacting deep neural networks for Internet of Things: Methods and applications," *IEEE Internet Things J.*, vol. 8, no. 15, pp. 11935–11959, Aug. 2021, doi: [10.1109/JIOT.2021.3063497](https://doi.org/10.1109/JIOT.2021.3063497).
- [94] S. Han, H. Mao, and W. J. Dally, "Deep compression: Compressing deep neural networks with pruning, trained quantization and Huffman coding," in *Proc. Int. Conf. Learn. Representations*, 2016, pp. 1–14.
- [95] A. Polyak and L. Wolf, "Channel-level acceleration of deep face representations," *IEEE Access*, vol. 3, pp. 2163–2175, 2015, doi: [10.1109/ACCESS.2015.2494536](https://doi.org/10.1109/ACCESS.2015.2494536).
- [96] S. Han et al., "Learning both weights and connections for efficient neural network," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, Art. no. 28.
- [97] R. Yu et al., "NISP: Pruning networks using neuron importance score propagation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 9194–9203.
- [98] H. Zhou, J. M. Alvarez, and F. Porikli, "Less is more: Towards compact CNNs," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 662–677.
- [99] W. Wen, C. Wu, Y. Wang, Y. Chen, and H. Li, "Learning structured sparsity in deep neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, D. D. Lee, M. Sugiyama, and U. V. Luxburg, Eds., 2016, vol. 29, pp. 2082–2090.
- [100] H. Li et al., "Pruning filters for efficient ConvNets," *CoRR*, 2016, pp. 2074–2082.
- [101] M. Courbariaux et al., "Binarized neural networks: Training deep neural networks with weights and activations constrained to +1 or -1," 2016, *arXiv:1602.02830*.
- [102] M. Rastegari et al., "Xnor-net: Imagenet classification using binary convolutional neural networks," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 525–542.
- [103] W. Li and J. Plested, "Pruning convolutional neural network with distinctiveness approach," in *Proc. Int. Conf. Neural Inf. Process.*, 2019, pp. 448–455.

- [104] Y. Gong et al., "Compressing deep convolutional networks using vector quantization," 2014, *arXiv:1412.6115*.
- [105] J. Wu, C. Leng, Y. Wang, Q. Hu, and J. Cheng, "Quantized convolutional neural networks for mobile devices," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 4820–4828.
- [106] M. Courbariaux, Y. Bengio, and J. David, "Binaryconnect: Training deep neural networks with binary weights during propagations," in *Proc. Adv. Neural Inf. Process. Syst.: Annu. Conf. Neural Inf. Process. Syst.*, 2015, pp. 3123–3131.
- [107] M. Rastegari et al., "Xnor-net: Imagenet classification using binary convolutional neural networks," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 525–542.
- [108] P. Merolla, R. Appuswamy, J. V. Arthur, S. K. Esser, and D. S. Modha, "Deep neural networks are robust to weight binarization and other non-linear distortions," 2016, *arXiv:1606.01981*.
- [109] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," 2015, *arXiv:1503.02531*.
- [110] J. H. Cho and B. Hariharan, "On the efficacy of knowledge distillation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 4794–4802.
- [111] X. Lan, X. Zhu, and S. Gong, "Knowledge distillation by on-the-fly native ensemble," in *Proc. 32nd Int. Conf. Neural Inf. Process. Syst.*, 2018, pp. 7528–7538.
- [112] R. Adriana et al., "Fitnets: Hints for thin deep nets," in *Proc. Int. Conf. Learn. Representatives*, vol. 2, 2015.
- [113] B. Peng et al., "Correlation congruence for knowledge distillation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 5007–5016.
- [114] T. G. Kolda et al., "Tensor decompositions and applications," in *Proc. SIAM Rev.*, 2009, pp. 455–500.
- [115] K. Hayashi et al., "Exploring unexplored tensor network decompositions for convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, Art. no. 32.
- [116] C. Tai et al., "Convolutional neural networks with low-rank regularization," 2015, *arXiv:1511.06067*.
- [117] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le, "Learning transferable architectures for scalable image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 8697–8710.
- [118] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.
- [119] C. Szegedy et al., "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 1–9.
- [120] 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.
- [121] 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. 5987–5995.
- [122] 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.
- [123] F. N. Iandola et al., "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and < 0.5 MB model size," 2016, *arXiv:1602.07360*.
- [124] A. Gholami et al., "Squeezenext: Hardware-aware neural network design," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2018, pp. 1719–171909.
- [125] A. G. Howard et al., "Mobilenets: Efficient convolutional neural networks for mobile vision applications," 2017, *arXiv:1704.04861*.
- [126] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 4510–4520.
- [127] A. Howard et al., "Searching for MobileNetV3," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 1314–1324.
- [128] M. Tan et al., "MnasNet: Platform-aware neural architecture search for mobile," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 2815–2823.
- [129] X. Zhang, X. Zhou, M. Lin, and J. Sun, "Shufflenet: An extremely efficient convolutional neural network for mobile devices," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 6848–6856.
- [130] N. Ma et al., "ShuffleNet V2: Practical guidelines for efficient CNN architecture design," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 116–131.
- [131] R. J. Wang, X. Li, and C. X. Ling, "Pelee: A real-time object detection system on mobile devices," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, Art. no. 31.
- [132] G. Ghiasi, T.-Y. Lin, and Q. V. Le, "NAS-FPN: Learning scalable feature pyramid architecture for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 7029–7038.
- [133] M. Abadi et al., "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," 2016, *arXiv:1603.04467*.
- [134] Y. Q. Jia et al., "Caffe: Convolutional architecture for fast feature embedding," in *Proc. 22nd ACM Int. Conf. Multimedia*, 2014, pp. 675–678.
- [135] M. Peemen, A. A. A. Setio, B. Mesman, and H. Corporaal, "Memory-centric accelerator design for convolutional neural networks," in *Proc. IEEE 31st Int. Conf. Comput. Des.*, 2013, pp. 13–19, doi: [10.1109/ICCD.2013.6657019](https://doi.org/10.1109/ICCD.2013.6657019).
- [136] C. Zhang et al., "Optimizing FPGA-based accelerator design for deep convolutional neural networks," in *Proc. ACM/SIGDA Int. Symp. Field-Programmable Gate Arrays*, 2015, pp. 161–170, doi: [10.1145/2684746.2689060](https://doi.org/10.1145/2684746.2689060).
- [137] T. Chen et al., "Dianna: A small-footprint high-throughput accelerator for ubiquitous machine-learning," *ACM SIGARCH Comput. Architecture News*, vol. 42, no. 1, pp. 269–284, 2014, doi: [10.1145/2654822.2541967](https://doi.org/10.1145/2654822.2541967).
- [138] Y. Chen et al., "Dadianna: A machine-learning supercomputer," in *Proc. 47th Annu. IEEE/ACM Int. Symp. Microarchitecture*, 2014, pp. 609–622, doi: [10.1109/MICRO.2014.58](https://doi.org/10.1109/MICRO.2014.58).
- [139] Y.-H. Chen, T. Krishna, J. S. Emer, and V. Sze, "Eyeriss: An energy-efficient reconfigurable accelerator for deep convolutional neural networks," *IEEE J. Solid-State Circuits*, vol. 52, no. 1, pp. 127–138, Jan. 2017, doi: [10.1109/JSSC.2016.2616357](https://doi.org/10.1109/JSSC.2016.2616357).
- [140] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 580–587.
- [141] 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, doi: [10.1109/TPAMI.2015.2389824](https://doi.org/10.1109/TPAMI.2015.2389824).
- [142] 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, doi: [10.1109/TPAMI.2016.2577031](https://doi.org/10.1109/TPAMI.2016.2577031).
- [143] J. Dai et al., "R-FCN: Object detection via region-based fully convolutional networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, Art. no. 29.
- [144] Z. Li et al., "Light-head R-CNN: In defense of two-stage object detector," 2017, *arXiv:1711.07264*.
- [145] 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, doi: [10.1109/CVPR.2016.91](https://doi.org/10.1109/CVPR.2016.91).
- [146] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 6517–6525, doi: [10.1109/CVPR.2017.690](https://doi.org/10.1109/CVPR.2017.690).
- [147] A. Farhadi and J. Redmon, "Yolov3: An incremental improvement," in *Computer Vision and Pattern Recognition*, vol. 1804, Berlin/Heidelberg, Germany: Springer, 2018, pp. 1–6.
- [148] A. Bochkovskiy, C. Y. Wang, and H. Y. M. Liao, "Yolov4: Optimal speed and accuracy of object detection," 2020, *arXiv:2004.10934*.
- [149] 2019. [Online]. Available: <https://github.com/ultralytics/yolov5>
- [150] Z. Ge et al., "Yolox: Exceeding yolo series in 2021," 2021, *arXiv:2107.08430*.
- [151] 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. 936–944.
- [152] 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, doi: [10.1109/CVPR.2018.00644](https://doi.org/10.1109/CVPR.2018.00644).
- [153] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 2, pp. 386–397, Feb. 2020.
- [154] S. Liu, D. Huang, and Y. Wang, "Learning spatial fusion for single-shot object detection," 2019, *arXiv:1911.09516*.
- [155] W. Liu et al., "SSD: Single shot multibox detector," in *Proc. Comput. Vis. – Eur. Conf. Comput. Vis.*, 2016, pp. 21–37, doi: [10.1007/978-3-319-46448-0\\_2](https://doi.org/10.1007/978-3-319-46448-0_2).
- [156] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 2, pp. 318–327, Feb. 2020, doi: [10.1109/TPAMI.2018.2858826](https://doi.org/10.1109/TPAMI.2018.2858826).
- [157] H. Law and J. Deng, "CornerNet: Detecting objects as paired keypoints," *Int. J. Comput. Vis.*, vol. 128, pp. 642–656, 2020, doi: [10.1007/s11263-019-01204-1](https://doi.org/10.1007/s11263-019-01204-1).

- [158] X. Long et al., “PP-YOLO: An effective and efficient implementation of object detector,” 2020, *arXiv:2007.12099*.
- [159] X. Huang et al., “PP-YOLOv2: A practical object detector,” 2021, *arXiv:2104.10419*.
- [160] G. Yu et al., “PP-PicoDet: A better real-time object detector on mobile devices,” 2021, *arXiv:2111.00902*.
- [161] K. Duan, S. Bai, L. Xie, H. Qi, Q. Huang, and Q. Tian, “CenterNet: Keypoint triplets for object detection,” in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 6568–6577, doi: [10.48550/arXiv.1904.08189](https://doi.org/10.48550/arXiv.1904.08189).
- [162] Z. Tian, C. Shen, H. Chen, and T. He, “FCOS: Fully convolutional one-stage object detection,” in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 9626–9635, doi: [10.1109/ICCV.2019.00972](https://doi.org/10.1109/ICCV.2019.00972).
- [163] T. Kong et al., “FoveaBox: Beyond anchor-based object detector,” *IEEE Tran. Image Process.*, vol. 29, pp. 7389–7398, 2020.
- [164] Z. Shen, Z. Liu, J. Li, Y.-G. Jiang, Y. Chen, and X. Xue, “Object detection from scratch with deep supervision,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 2, pp. 398–412, Feb. 2020, doi: [10.1109/TPAMI.2019.2922181](https://doi.org/10.1109/TPAMI.2019.2922181).
- [165] Z. Shen et al., “Learning object detectors from scratch with gated recurrent feature pyramids,” 2017, *arXiv:1712.00886*.
- [166] Z. Li et al., “DetNet: A backbone network for object detection,” 2018, *arXiv:1804.06215*.
- [167] R. Zhu et al., “ScratchDet: Training single-shot object detectors from scratch,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 2263–2272, doi: [10.1109/CVPR.2019.00237](https://doi.org/10.1109/CVPR.2019.00237).
- [168] K. He, R. Girshick, and P. Dollár, “Rethinking imagenet pre-training,” in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 4917–4926, doi: [10.1109/ICCV.2019.00502](https://doi.org/10.1109/ICCV.2019.00502).
- [169] A. Banitalebi-Dehkordi, “Revisiting knowledge distillation for object detection,” 2021, *arXiv:2105.10633*.
- [170] Q. Li, S. Jin, and J. Yan, “Mimicking very efficient network for object detection,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 7341–7349.
- [171] T. Wang, L. Yuan, X. Zhang, and J. Feng, “Distilling object detectors with fine-grained feature imitation,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 4928–4937.
- [172] A. Chawla, H. Yin, P. Molchanov, and J. Alvarez, “Data-free knowledge distillation for object detection,” in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis.*, 2021, pp. 3288–3297.
- [173] G. Chen et al., “Learning efficient object detection models with knowledge distillation,” in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 742–751.
- [174] Y. Wang et al., “A SAR dataset of ship detection for deep learning under complex backgrounds,” *Remote Sens.*, vol. 11, no. 7, 2019, Art. no. 765, doi: [10.3390/rs11070765](https://doi.org/10.3390/rs11070765).
- [175] S. Xian et al., “AIR-SARShip-1.0: High-resolution SAR ship detection dataset,” *J. Radars*, vol. 8, no. 6, pp. 852–862, 2019.
- [176] 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, doi: [10.3390/rs12122031](https://doi.org/10.3390/rs12122031).
- [177] 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, doi: [10.3390/rs12182997](https://doi.org/10.3390/rs12182997).
- [178] 2020. [Online]. Available: <http://radars.ie.ac.cn/web/data/getData?newsColumnId=74fe223a-0b01-4830-8d99-1ba276e67ad8&pageType=en>
- [179] C. Ding, “SRSD-V1.0: A high-resolution SAR rotation ship detection dataset,” *Remote Sens.*, vol. 13, no. 24, 2021, Art. no. 5104, doi: [10.3390/rs13245104](https://doi.org/10.3390/rs13245104).
- [180] X. Congan et al., “RSDD-SAR: Rotated ship detection dataset in SAR images,” *J. Radars*, vol. 11, no. 4, pp. 581–599, 2022, doi: [10.12000/JR22007](https://doi.org/10.12000/JR22007).



**Jianwei Li** was born in 1989. He received the bachelor's and master's degrees in engineering from the Aviation University of Air Force, Changchun, China, in 2012 and 2015, respectively, and the Ph.D. degree in engineering from the Naval Aviation University, Yantai, China, in 2018.

He is currently an Assistant Researcher with the Navy Submarine Academy. His research interests include electronic countermeasure, SAR image processing, deep learning, and computer vision.



**Jie Chen** was born in 1980. She received the bachelor's, master's, and Ph.D. degrees in engineering from the Rocket Army Engineering University, Xi'an, China, in 2003, 2006, and 2010 respectively.

She is currently an Associate Researcher with the Navy Submarine Academy. Her research interests include ocean remote sensing, remote sensing image detection, and recognition.



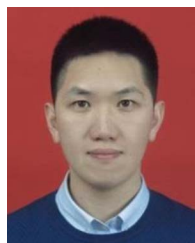
**Pu Cheng** was born in 1988. He received the Ph.D. degree in engineering from the National University of Defense Technology, Changsha, China, in 2017.

He is currently a Lecturer with the Navy Submarine Academy. His research interests include ocean remote sensing, radar signal processing, and synthetic aperture radar.



**Zhentao Yu** was born in 1984. He received the Ph.D. degree in engineering from the Naval Aviation University, Yantai, China, in 2014.

He is currently a Researcher with the Naval Submarine Academy. His research interests include image processing and magnetic target detection target recognition.



**Lu Yu** received the B.S. and M.S. degrees in engineering from the Naval University of Engineering, Wuhan, China, in 2012 and 2015, respectively, and the Ph.D. degree in engineering from the Naval Aviation University, Yantai, China, in 2018.

He is currently a Lecturer with the Naval Submarine Academy. His research interests include deep learning, computer vision, and remote sensing interpretation.



**Cheng Chi** received the B.S. degree in coastal engineering from the HoHai University, Nanjing, China, in 2011, the M.S. degree in physical oceanography from the Navy Submarine Academy, Qingdao, China, in 2013, and the Ph.D. degree in control science and engineering from the Naval Aviation University, Yantai, China, in 2017.

He is currently a Lecturer with the Navy Submarine Academy, Qingdao, China. His research interests include signal processing and magnetic targets localization.