



Article

On-Board Real-Time Ship Detection in HISEA-1 SAR Images Based on CFAR and Lightweight Deep Learning

Pan Xu ^{1,2,3} , Qingyang Li ^{1,2,3}, Bo Zhang ⁴, Fan Wu ⁴, Ke Zhao ^{1,2,3}, Xin Du ^{1,2,3}, Cankun Yang ^{1,2,3} and Ruofei Zhong ^{1,2,3,*}

- ¹ Key Laboratory of 3D Information Acquisition and Application Ministry of Education, Capital Normal University, Beijing 100048, China; 2180902091@cnu.edu.cn (P.X.); 2200901013@cnu.edu.cn (Q.L.); 2200902154@cnu.edu.cn (K.Z.); 2200902172@cnu.edu.cn (X.D.); yangck@cnu.edu.cn (C.Y.)
- ² College of Resource Environment and Tourism, Capital Normal University, Beijing 100048, China
- ³ Beijing Advanced Innovation Center for Imaging Theory and Technology, Capital Normal University, Beijing 100048, China
- ⁴ The Key Laboratory of Digital Earth Sciences, Aerospace Information Research Institute, Chinese Academy of Sciences, Beijing 100094, China; zhangbo202140@aircas.ac.cn (B.Z.); wufan@aircas.ac.cn (F.W.)
- * Correspondence: zrf@cnu.edu.cn

Abstract: Synthetic aperture radar (SAR) satellites produce large quantities of remote sensing images that are unaffected by weather conditions and, therefore, widely used in marine surveillance. However, because of the hysteresis of satellite-ground communication and the massive quantity of remote sensing images, rapid analysis is not possible and real-time information for emergency situations is restricted. To solve this problem, this paper proposes an on-board ship detection scheme that is based on the traditional constant false alarm rate (CFAR) method and lightweight deep learning. This scheme can be used by the SAR satellite on-board computing platform to achieve near real-time image processing and data transmission. First, we use CFAR to conduct the initial ship detection and then apply the You Only Look Once version 4 (YOLOv4) method to obtain more accurate final results. We built a ground verification system to assess the feasibility of our scheme. With the help of the embedded Graphic Processing Unit (GPU) with high integration, our method achieved 85.9% precision for the experimental data, and the experimental results showed that the processing time was nearly half that required by traditional methods.

Keywords: ship detection; lightweight deep learning; CFAR algorithm; on-board processing



Citation: Xu, P.; Li, Q.; Zhang, B.; Wu, F.; Zhao, K.; Du, X.; Yang, C.; Zhong, R. On-Board Real-Time Ship Detection in HISEA-1 SAR Images Based on CFAR and Lightweight Deep Learning. *Remote Sens.* **2021**, *13*, 1995. <https://doi.org/10.3390/rs13101995>

Academic Editor: Alin Achim

Received: 24 March 2021

Accepted: 13 May 2021

Published: 19 May 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Unlike optical satellites, synthetic aperture radar (SAR) satellites work in all weather conditions thanks to active data collection. They are widely used in marine monitoring and are the main remote sensing means for large-scale maritime ship target detection. Marine ship target detection can monitor maritime traffic, maintain maritime rights and interests, and improve the early warning capabilities of coastal defenses [1–3]. Therefore, high precision positioning of ship targets is required. The traditional image information acquisition process involves four main steps: data acquisition, data download, ground centralized processing, and information extraction. This process can be slow, and the longer the time from when the satellite generates the data to when information is extracted on the ground, the less useful that data will be. It is necessary to migrate the processing and information extraction algorithms from the ground to the on-board computing platform in order to make full use of the limited transmission bandwidth and satellite transmission time to shorten the time delay in information acquisition. This migration would also reduce the load on the satellite-ground data transmission system [4–6].

Many satellites have the capability of on-board processing of remote sensing data. The Earth Observation-1 launched by NASA's New Millennium Program in November of 2000 has on-board processing capabilities that include emergency monitoring, feature monitoring, change monitoring, and anomaly monitoring. In October 2001, DigitalGlobe launched the QuickBird satellite, which was the highest resolution commercial satellite to date and it can produce multiple types of images and perform remote sensing image preprocessing and real-time multispectral classification on-board. The NEMO satellite, launched by the US, is equipped with COIS coast imaging spectrometer and adaptive spectrum identification system ORASIS and it can generate and directly download coastal description information [7]. On-board processing is usually implemented by Digital Signal Processor and Field-Programmable Gate Array; however, due to their poor scalability, implementing new algorithms is difficult, which restricts them from meeting the diverse needs of intelligent on-board processing. However, with rapid advancements in computing hardware, such as the advent of embedded Graphic Processing Units (GPUs) with low power consumption, strong performance, and high integration, new solutions for on-board real-time processing have been made available [8], such as NVIDIA Jetson TX1 and TX2.

In the past few years, researchers have proposed many traditional ship detection methods utilizing SAR imagery [9–12]. Gao et al. [9] proposed an adaptive and fast CFAR algorithm that was based on automatic censoring (AC), and the combination of AC with the algorithm resulted in good performance. Huang et al. [10] used a new ship detection approach based on multi-scale heterogeneities under the a contrario decision framework, this method proved to be suitable for ships of different sizes. Ji et al. [11] used the K-distribution CFAR method to calculate a global threshold with a given false-alarm rate, then mathematical morphological filters are applied to the results; this approach greatly improved the processing speed. Ai et al. [12] proposed an algorithm for ship target gray distributions based on correlations, and then the joint gray distribution models of pixels and adjacent pixels in clutter are established using the two-dimensional joint lognormal distribution. However, these methods are dependent on the pre-classification and pre-definition of features by humans, and they may be less robust in complex backgrounds. Furthermore, no on-board traditional method for real-time SAR processing has been verified.

With the development of computer vision, deep learning has gradually been applied to ship detection from SAR imagery. Target detectors that are based on deep learning can be divided into two categories. The first type of target detectors are based on region recommendations, such as Region-based Convolutional Networks (R-CNN) [13], Spatial Pyramid Pooling Network (SPP-NET) [14], Fast R-CNN [15], and Faster R-CNN [16]. This type of detector first uses regional recommendations to generate candidate targets, and then uses the convolution neural network for processing. Its accuracy is high, but it does not meet the requirements for real-time processing. The second type of target detectors are regression-based, such as You Only Look Once (YOLO) [17–19], Single Shot Multibox Detector (SSD) [20], etc. These methods treat the detection problem as a regression and directly predict target positions and categories. This type of detector has a fast processing speed, but low accuracy. Cui et al. [2] applied R-CNN to large scene SAR target recognition, which can detect objects while recognizing their classes thanks to its regression method and the sharing network structure. Wang et al. [21] used RetinaNet with feature pyramid networks (FPN) and focal loss to detect ships, and achieved high accuracy multi-scale ship detection. Kang et al. [22] used the Faster R-CNN method to conduct the initial ship detection and then applied CFAR to obtain the final results.

The challenge of on-board ship detection of lightweight SAR satellite is to achieve the accurate and efficient detection of targets under the constraints of the limited memory and computing power of the satellite processing platform. Most of the ship detection methods use high-power GPUs on the ground to detect ships, and these network models are always large in size and complex computations. However, the special hardware used for lightweight satellites is different from the mature technology available in ground data

centers. Therefore, these highly complex network models cannot be used to complete real-time detection missions with the limited on-board memory and computing power.

To solve the above problem, we propose an on-board ship detection scheme that combines the traditional CFAR algorithm and lightweight deep learning to achieve efficient on-board ship detection. Firstly, this method is based on geographic prior experience and regional growth methods to realize sea-land segmentation. Sea-land segmentation can reduce the interference of the land and improve detection efficiency. Secondly, the K-distribution is used to model sea clutter in images, and the CFAR algorithm is used for fast rough detection. Subsequently, the initial detection results are used as input for the YOLOv4-tiny network to refine the detection results. Finally, through a reasonable hardware acceleration strategy, the entire algorithm was ported into a Jetson TX1 GPU and achieved an acceptable algorithm speed. This method only uses the preliminary extraction results as the input for the YOLOv4-tiny network, which greatly reduces the computation power required. The main difference between our method and the vast majority of the SAR automatic target recognition approaches is that we consider an end-to-end workflow, in which manually extracting samples for target classification is not required. In addition, our algorithm can directly generate the category probability and position coordinate value of the object, and obtain the final detection result after a single detection, so it has a faster detection speed. When combined with the hardware acceleration strategy, our method achieved high accuracy and efficiency on the ground verification platform. The main contributions of this paper are as follows:

1. In order to eliminate the influence of land, we first remove the land from the image, which greatly reduces the computation load of the algorithm. Subsequently, we use the CFAR algorithm that is based on K-distribution to extract ship targets.
2. We input the CFAR detection results into the YOLOv4-tiny network for further ship detection using its efficient global context information extraction capability.
3. We combine each part of the algorithm organically in the Jetson TX1 platform. Through a reasonable hardware acceleration strategy, we reduce the running time of our algorithm. Additionally, our method was validated on the HISEA-1 satellite.

The organization of this paper is as follows. Section 2 relates to the details of our proposed ship detection algorithm, including the CFAR algorithm and YOLOv4-tiny. Section 3 introduces the satellite computing platform and the construction of the ground verification system. Section 4 reports on the experiments. Sections 5 and 6 present our discussion and conclusion.

2. Methods of On-Board Ship Detection

Because of the limitations of the GPU's memory, the original large-scale SAR remote sensing images should be broken into image patches before target detection. Because of the large-size of the original image, the number of image patches will be relatively large, which will not only take up the limited storage, but it will also result in the model processing many empty image blocks. This is why this paper proposed an on-board ship detection scheme based on the CFAR algorithm and deep learning. Using the CFAR method before deep learning, target recognition can significantly reduce the number of image patches submitted to deep learning detection. Because of the high complexity of the deep learning algorithm, the processing time will increase with an increased number of target patches. Therefore, the first extracting targets using CFAR will greatly reduce the computations that are required by deep learning. Our experimental results showed that using the CFAR method before deep learning reduced the computational time by more than half when compared to using only the deep learning method. First, the image is categorized as either sea or land, and if it includes the coast, it will be divided by sea and land. This is done to prevent land features from affecting the detection of ships. Subsequently, the CFAR algorithm is used for an initial quick and rough ship detection, and any detected targets are stored as 256×256 pixels image patches. The resulting

image patches are input into the trained and lightweight deep learning model for further detection to obtain a more accurate final ship detection. Finally, the targets are mapped on the original image to obtain its pixel coordinates, the latitude and longitude coordinates of the four corners of the original image are used to determine the targets actual location using the bilinear interpolation method, and then this effective information is transmitted through the satellite-earth data transmission system. Figure 1 shows the process of on-board ship detection.

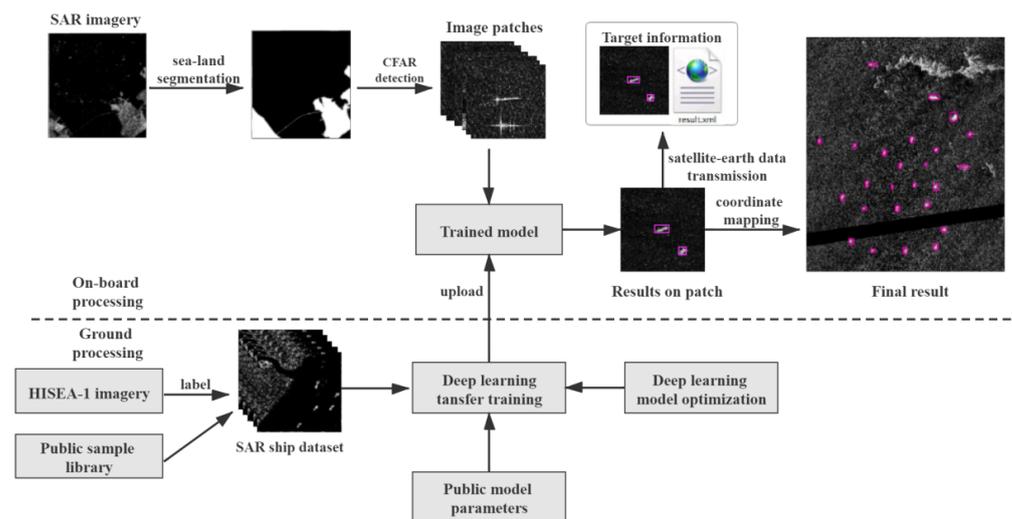


Figure 1. The process of on-board ship detection based on constant false alarm rate (CFAR) and deep learning.

2.1. Sea-Land Segmentation

During ship detection, land area is not of interest, but the SAR images that are used commonly include both land and sea. The purpose of sea-land segmentation is to mask or remove the land area from the SAR images, so that the detection algorithm only processes the sea area and ignores the land [23]. This is particularly important, because many land objects appear in SAR images as strong scatterers that are similar to ships at sea. These land objects will produce false alarms and waste processing power that should be focused on the detection of ships. Therefore, effective sea-land segmentation can limit the detection range to sea areas, thereby improving the detection efficiency and accuracy.

This paper first adopted a method in which geographic prior information (coastline vector data) is superimposed on SAR images using a vector layer, and the polygon vector elements in the vector layer are then used to determine whether the image is of the sea, land, or sea-land junction. The region growing method is then used for sea-land segmentation.

2.1.1. Judgment on Land and Sea

In this part, using the latitudes and longitudes of the four corners of the image, the global coastline vector data are superimposed to determine the position of the image. Because every polygon on the vector layer corresponds to a piece of land, the problem of judging whether corner points of the SAR image are sea or land is converted to a problem of whether the points on the vector layer are in the polygons designated as land. The problem of determining whether a point is in a polygon in the vector layer is carried out according to the graphics algorithm [24]; where, the time complexity of the algorithm is $O(n)$ and n is the number of polygonal line segments. Because of the large size of the acquired images and the requirements for efficiency, only the four corner points of the image are used to obtain the approximate position of the image.

2.1.2. Sea-Land Segmentation Based on Region Growth

Region growth is an image segmentation method that is based on image gray-scale similarity, neighborhood information, etc., with the basic goal of grouping pixels with similar properties to form regions [25]. Segmentation that is based on region growth has two key elements: one is to select the seed point of the image to be segmented and the other is to establish the range of pixel intensity contained in the region. Once there is a seed and a range, the area will grow from the seed point. When there are no more pixels that meet the conditions, the algorithm stops. T is the given threshold, so the seed growth criterion is expressed by Equation (1), as:

$$|y(i, j) - \mu_R| < T. \quad (1)$$

where, $y(i, j)$ is the gray value of the pixel to be determined; and, μ_R is the average intensity of the growth area R of the seed pixel.

The selection of seed points is based on the results of the land and sea judgment. If a corner point is judged to be land, then that pixel point is used as the seed for regional growth. Compared with the ocean area, the land area in the SAR image exhibits strong scattering and it has a higher gray value. The gray value is used as the threshold value and the sea and land area can then be separated through regional growth. Figure 2 shows examples of sea-land segmentation.

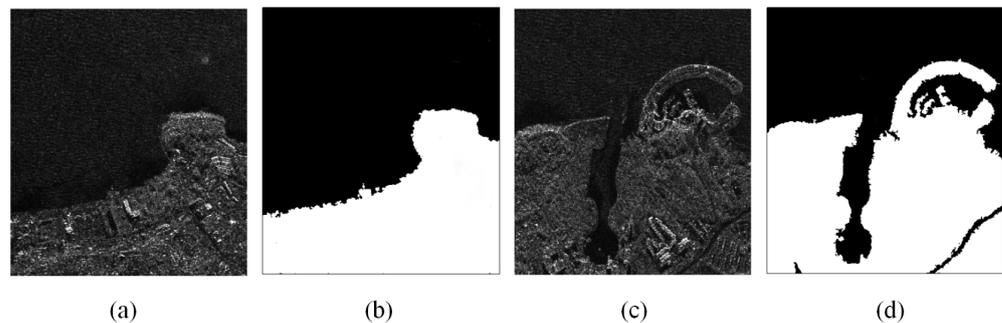


Figure 2. Representative examples of sea-land segmentation. (a,c) Original images. (b,d) Images after sea-land segmentation.

2.2. CFAR Detector Based on K-Distribution

When considering the large scale of the original image, recognition speed using only the deep learning model will be relatively slow. Therefore, this paper proposed using CFAR to perform the initial ship detection, the results of which are then used as the input for the YOLO method to obtain the final, more accurate, detection results. The initial ship detection results are stored as sub-images that are suitable for neural network recognition, and then input into the YOLO model to obtain accurate detection results.

The false alarm rate is the probability that the detector will misjudge noise or other interference signals as valid target signals in a given unit of time. By setting a constant false alarm rate, stable target detection results can be obtained during signal detection. It is necessary to carry out statistical analyses of the characteristics of sea clutter to accurately model background sea clutter in order to obtain a constant false alarm effect under different sea and environmental conditions. For this reason, different clutter distribution models have been proposed, including the Rayleigh, Lognor-mal, Weibull, α -steady distribution, G^0 distribution [26], and K-distribution [27] models, among which the K-distribution is the most commonly used statistical model. It not only accurately matches the clutter amplitude distribution over a wide range, but also accurately simulates the correlations between clutter echo pulses. As such, it has become a classic model for describing sea clutter.

2.2.1. Parameter Estimation on K-Distribution Model

The probability statistical model of the K-distribution [28] is as follows:

$$p(x) = \frac{2}{a\Gamma(v)} \left(\frac{x}{2a}\right)^v K_{v-1}\left(\frac{x}{a}\right), \quad (2)$$

where, $\Gamma(\cdot)$ is the Gamma function; $K_{v-1}(\cdot)$ is the second type modified Bessel function of order $v-1$; a is the scale parameter; and, v is the shape parameter. For most sea clutter, the shape parameter values range is $0.1 < v < \infty$. When $v \rightarrow \infty$, the distribution of the clutter is close to the Rayleigh distribution.

Subsequently, we chose the statistical estimation method to estimate the parameters. From the K-distribution model expression, the r -moment of the K-distribution can be obtained, as shown in Equation (3):

$$\mu_r = E[x^r] = \frac{\Gamma(v + 0.5r)\Gamma(1 + 0.5r)}{\Gamma(v)} (2a)^r. \quad (3)$$

According to the properties of the Gamma function presented in Equation (4), we can obtain Equation (5).

$$\Gamma(x + 1) = x \cdot \Gamma(x), \quad (4)$$

$$z_r = \frac{\mu_{r+2}}{\mu_r} = (2 + r)a^2(2v + r). \quad (5)$$

It can be seen from Equation (5) that the moment estimation method only requires that multiple z_r groups be calculated. For sea clutter samples $\{x_i; i = 1, 2, \dots, N\}$, the k -order sample moment of the K-distribution is expressed by Equation (6):

$$\mu_k = \frac{1}{N} \sum_{i=1}^N x_i^k. \quad (6)$$

Random sample moments are used instead of overall moments and an inverse transformation is performed to obtain the scale parameter a and shape parameter v .

2.2.2. CFAR Ship Detection Based on the K-Distribution

The CFAR algorithm determines the threshold based on the statistically determined characteristics of the image pixels. The basic principle is that, for each pixel x_i , a certain reference window around it is defined and threshold I_c is determined based on the characteristics of the pixels in the window. Target detection can be achieved by comparing the value of the pixel with the value of the threshold. The sea background statistical model uses the K-distribution, and the specific process is as follows:

- (1) The scale parameter a and shape parameter v of the K-distribution are calculated using Equations (3)–(5).
- (2) The parameter estimations are used in Equation (1) to obtain the probability density function that is used to solve Equation (7) for I_c , as follows:

$$P_{fa} = 1 - \int_0^{I_c} p(x) dx, \quad (7)$$

where, P_{fa} is the false alarm rate.

- (3) According to the detection threshold I_c obtained by Equation (7), we can determine whether the pixel x_i to be detected in the target window is a target pixel based on Equation (8):

$$x_i > I_c, \quad (8)$$

if the equation is satisfied, then it is categorized as a target; otherwise, it is categorized as background.

2.3. Ship Detection Based on the YOLOv4-Tiny Network

Traditional deep learning models have the disadvantages of involving many parameters and being very large. Applications in various fields have shown that YOLO is better at generalization than R-CNN. We constructed a YOLOv4-tiny-based [29] training convolutional neural network to obtain the final ship detection results in order to achieve real-time and efficient detection of ship targets, while, at the same time, considering model size and subsequent model injection through the satellite-earth data transmission system. The model was only 22.4 MB, which would satisfy the mission requirements for on-board real-time detection.

2.3.1. Architecture of the YOLOv4-Tiny Network

YOLO is an algorithm that integrates target area and target category predictions into a single neural network. First, the images are divided into grids with sizes $S \times S$, where each grid cell is responsible for the detection of the object that is centered in that cell. The predicted confidence scores indicate the accuracy of the prediction results for each grid cell. If there is no object in the cell, the confidence score is zero. Otherwise, the confidence score is equal to the intersection over union (IoU) between the ground truth and prediction box [18]. We used a confidence score threshold to determine whether the predicted bounding boxes should be retained.

The CSPDarknet53-tiny network is a simplified version of CSPDarknet53, and it uses the CSPBlock module instead of the ResBlock module. The CSPBlock module can enhance the learning ability of convolution network comparing with ResBlock module. At the same time, it improves the inference speed and accuracy of the YOLOv4-tiny method.

The building blocks of CSPDarknet53 can be divided into two groups: the convolutional building block and the five CSPBlock modules. The convolutional building block contains a convolutional (Conv) layer with a kernel size of 3, a stride of 1, and a Mish layer. The CSPBlocks divide the feature map of the base layer into two parts, and then merges them through a cross-phase hierarchy, as shown in Figure 3a. In this way, the gradient flow can propagate through different network paths after being separated [30]. The five CSPBlock modules are repeatedly stacked by the two Conv layers in the red box in Figure 3. The numbers of stacked Conv layers are 1, 2, 8, 8, and 4, respectively. The first Conv layer has a kernel size of 3 and a stride of 2, and the second Conv layer has a kernel size of 3, and a stride of 1. YOLOv4-tiny is designed based on YOLOv4, but it prioritizes faster object detection speeds. YOLOv4-tiny uses the CSPDarknet53-tiny network as its backbone network, it's network structure is shown in Figure 4.

CSPDarknet53-tiny consists of three Conv layers and three CSPBlock modules. Specifically, there is a Conv layer and LeakyReLU layer in the first two Conv layers, and each Conv layer has a kernel size of 3 and a stride of 2. The last Conv layer has kernel size of 3 and a stride of 1. CSPDarknet53-tiny uses the LeakyReLU function as the activation function in place of the Mish activation function, which simplifies the calculation process.

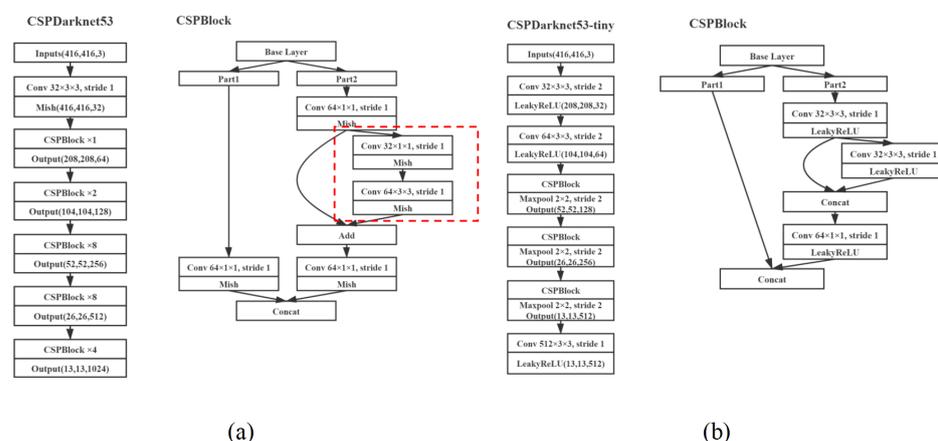


Figure 3. The structure of CSPDarknet53 (a) and CSPDarknet53-tiny (b).

The YOLOv4-tiny uses two different scales feature maps that are 13×13 and 26×26 to predict the detection results. At the same time, each feature map uses three anchor boxes of different sizes to predict three bounding boxes, and k-means clustering is used in advance to pre-define the number of anchor boxes, and there is a total of six pre-defined boxes for the three scales.

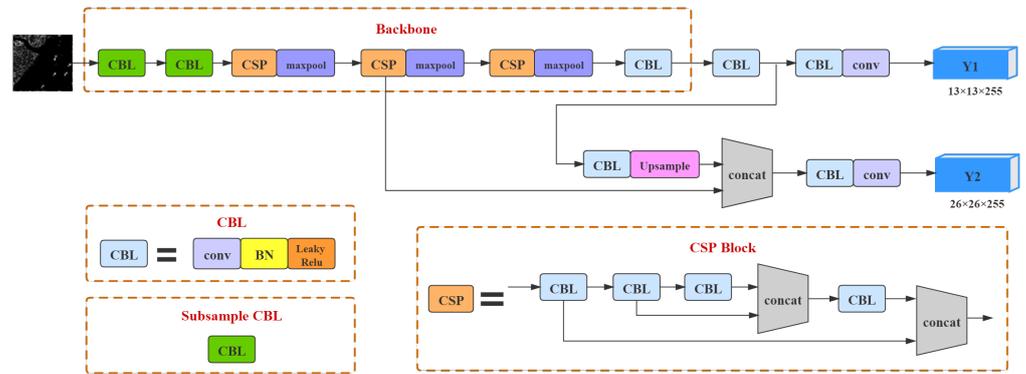


Figure 4. YOLOv4-tiny network architecture.

2.3.2. Loss Function

The loss function of YOLOv4-tiny contains three parts and can be expressed as follows:

$$loss = l_{box} + l_{obj} + l_{cls}, \quad (9)$$

where, l_{box} , l_{obj} , and l_{cls} are confidence bounding box regression loss, classification loss, and confidence loss functions, respectively.

YOLOv4-tiny uses complete intersection over union (CIoU) loss instead of Mean Square Error (MSE) loss, which takes the three geometric factors of the two detection boxes into account, namely their overlapping area, the distance between their center points, and their aspect ratios. Accordingly, the bounding box regression loss function is:

$$l_{box} = 1 - IoU + \frac{\rho^2(b, b^{gt})}{c^2} + \alpha \cdot v, \quad (10)$$

$$\alpha = \frac{v}{(1 - IoU) + v}, \quad (11)$$

$$v = \frac{4}{\pi^2} \left(\arctan \frac{w^{gt}}{h^{gt}} - \arctan \frac{w}{h} \right)^2, \quad (12)$$

where, IoU is the intersection over union between the truth bounding box and predicted bounding box; b is the coordinate of the center point of the predicted bounding box; b^{gt} is the center point coordinates of the real bounding box; $\rho(\cdot)$ is the calculation of Euclidean distance; c is the diagonal distance of the box that can contain the predicted and truth bounding boxes; w^{gt} and h^{gt} are the width and height of the truth bounding box; and, w and h are the width and height of the predicted bounding box.

The confidence loss function is:

$$l_{obj} = - \sum_{i=0}^{S \times S} \sum_{j=0}^B I_{ij}^{obj} [C_i \log(\hat{C}_i) + (1 - C_i) \log(1 - \hat{C}_i)] - \sum_{i=0}^{S \times S} \sum_{j=0}^B (1 - I_{ij}^{obj}) [C_i \log(\hat{C}_i) + (1 - C_i) \log(1 - \hat{C}_i)], \quad (13)$$

where, $S \times S$ is the grid size; B is the number of bounding boxes in a grid; I_{ij}^{obj} is a function of the object, if the j th bounding box of the i th grid is responsible for detecting the current

object, $I_{ij}^{obj} = 1$, otherwise $I_{ij}^{obj} = 0$; \hat{C}_i and C_i are the confidence scores of the predicted and truth boxes, respectively.

The classification loss function is:

$$lcls = - \sum_{i=0}^{S \times S} \sum_{j=0}^B I_{ij}^{obj} \sum_{c \in \text{classes}} [p_i(c) \log(\hat{p}_i(c)) + (1 - p_i(c)) \log(1 - \hat{p}_i(c))], \quad (14)$$

where, $\hat{p}_i(c)$ and $p_i(c)$ are the predicted probability and true probability that the object belongs to c classification in the j th bounding box of the i th grid.

The detection results will produce a large number of overlapping detection bounding boxes. Based on traditional Non-Maximum Suppression (NMS), distance intersection over union non-maximum suppression (DIoU-NMS) is used to resolve problems when the bounding boxes do not overlap, using the distances between the center points of the bounding boxes to do so. We chose 0.4 as the threshold for DIoU-NMS in this article.

3. Satellite Ground Verification System and Algorithm Hardware Acceleration Strategy

3.1. Algorithm Hardware Adaptation and Acceleration Strategy

We used the Jetson TX1 as the on-board development board in order to realize on-board ship detection. The embedded vision computer system Jetson TX1, which was developed by NVIDIA integrates an ARM Cortex-A57 Central Processing Unit (CPU) and a 256-core NVIDIA Maxwell architecture GPU, can realize trillions of floating-point operations per second on Linux systems and is very suitable for artificial intelligence calculations. At the same time, its small size, low power consumption, and high integration make it suitable for real-time on-board satellite processing.

In order to give full play to the computing power of the Jetson TX1 hardware, we fully analyzed the algorithm principles and assigned each module to the appropriate hardware to maximize the calculation efficiency. Specifically, due to the large number of logic operations required by the sea-land segmentation and CFAR algorithms, which belong to the serial logic flow algorithm, they are designated to the CPU, which has superior serial computing power. For the YOLOv4-tiny network, a large number of matrix convolution operations are involved with floating point type data. For these operations, the GPU can provide better throughput, albeit with the cost of a delay. Taking full advantage of the multi-core, the GPU is much better than the CPU in parallel computing and floating-point computing efficiency. Therefore, we assigned the YOLOv4-tiny network to the GPU.

In addition, in order to prevent the time taken to transfer data between the CPU and GPU from affecting the overall performance of the program, we made the calculation workload of the GPU far greater than the workload of copying and transferring data so that it does not copy and transfer frequently. This can reduce the proportion of time that is spent copying and transmitting data relative to the total time consumed, which will reduce its effect on overall performance. At the same time, when multiple threads are ongoing on the CPU, one thread can be in the process of copying while the other thread is performing calculations. In this way, calculations and copying can be performed simultaneously to reduce the calculation time. Finally, using the stream mechanism of the CUDA GPU, in asynchronous mode, when one stream is copying the other is calculating.

3.2. Composition of Ground Verification System

We constructed a ground verification system for testing before porting it into the satellite to verify the feasibility of our scheme. All of our experiments were carried out in this simulated environment. Figure 5 shows the system schematic.

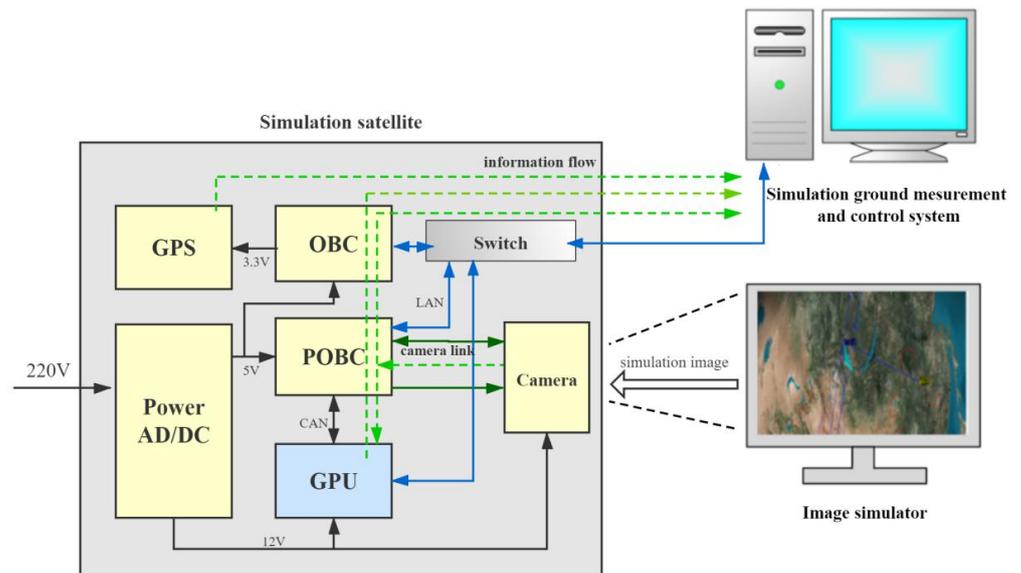


Figure 5. Schematic of the ground verification system.

The main components of the system were the OBC (on-board computer), POBC (on-board data processing computer), GPS (global positioning system) module, switch module, camera, image simulator, and simulated ground measurement and control system. As the control center of the system, the OBC runs the satellite service software and controls the entire satellite. It sends the status and telemetry information from the satellite to the simulated ground measurement and control system, and it communicates with the GPS module through the serial port. The POBC receives the camera's image data through the camera link, processes it as required, and then transmits it via ethernet. The simulated ground measurement and control system displays the satellite telemetry signal and status parameters. The GPS module provides latitude and longitude information.

With this system, the information flows from the camera, which collects images of the image simulator, to the POBC through the camera link for processing and storage, before finally travelling to the simulated ground measurement and control system where the data are displayed.

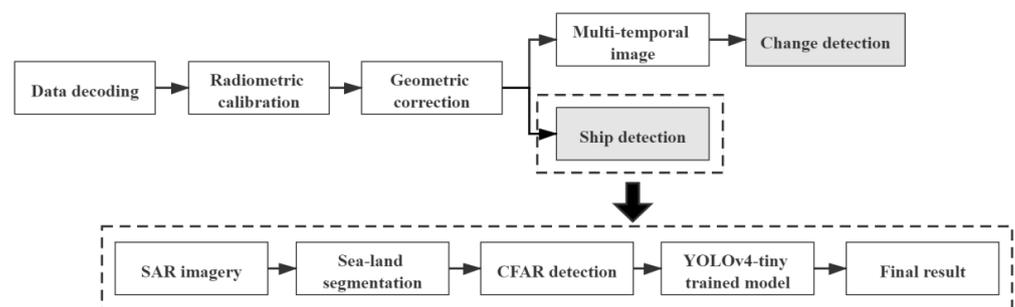
3.3. The HISEA-1 Satellite and Its On-Board Image Processing Load

At 12:37 noon Beijing time on December 22, 2020, the HISEA-1 satellite that was developed by Spacety was successfully launched on the CZ-8 rocket. The HISEA-1 satellite represented the first SAR remote sensing satellite constructed for scientific observation of oceans and coastal zones by a domestic university. It is also the first star of the planned HISEA series of small satellite constellations. The constellation will include multiple small and light SAR satellites and multi-spectral satellites that can be utilized by various field observation systems. Table 1 lists the requirements for the HISEA-1 mission. On 25 December 2020, the first batch of images from the HISEA-1 satellite were successfully obtained. To reduce the pressure of data transmission, the satellite platform is equipped with an intelligent information processing module for technical exploration of on-board intelligent image processing. This module is designed to extract effective information from remote sensing images and, thereby, reduce the data load sent to the measurement and control station and improve the efficiency of data acquisition.

Table 1. HISEA-1 mission parameters.

Index	Value
Orbit	400–500 km
Center frequency	5.4 GHz
Bandwidth	60–300 MHz
Incidence angle range	20°–35°
Polarization	VV, HH
Resolution and swath	3 m @ 20 km, 20 m @ 100 km, 10 m @ 50 km, 1 m @ 5 km * 5 km
Positional accuracy	≤200 m
Absolute radiometric accuracy	2.0 dB
Antenna size	4.0 m (A) × 0.64 m (R)
Average power	<2500 W
Total mass	185 kg

The on-board image processing load of the HISEA-1 satellite includes two parts, one is the SAR image preprocessing module and the other is an intelligent application module. Pre-processing includes data decoding, radiometric calibration, and geometric correction. The intelligent application module is responsible for real-time ship detection and detection of ground changes. Figure 6 shows the processing flow. Using the pre-processed images, the on-board ship detection process will extract the target image patches and position information, producing the effective information that will be transmitted to the ground receiver through the downlink. At the same time, the ground control system can monitor satellite transmission data in real time and communicate ground instructions to the satellite and, by continually updating the algorithm model parameters in the satellite's intelligent processing unit through the satellite upload channel, it can be tasked to complete more on-board processing tasks.

**Figure 6.** On-board intelligent processing platform processing flow.

4. Experimental Results and Analysis

In this section, we will explain the datasets used for the experiment, the process of the experiment, and discuss the results. First, the dataset was divided into three parts, i.e., the training set, validation set, and the testing set with their respective proportions being 7:2:1. We applied different versions of the YOLO model, SSD, and Faster R-CNN to train the dataset and compared the results. Next, we used the Sentinel-1 simulated images for ship detection experiments and evaluated the accuracy of the results. Finally, we tested our proposed method on real images from HISEA-1.

4.1. Experimental Dataset

In this work, we used the public SAR image ship detection dataset from the Key Laboratory of Digital Earth Sciences, Aerospace Information Research Institute, Chinese Academy of Sciences [31], which was labeled by Wang et al, part of the sample images is shown in Figure 7. It contained 102 Gaofen-3 images and 108 Sentinel-1 images, in which there were 59,535 ships in 43,819 ship chips. The images varied in terms of polarization, resolution, incidence angle, imaging mode, and background. The dataset was created for vari-

ous SAR image applications and the size of each image in the dataset was 256×256 pixels. In this dataset, each image target in the dataset has a ground truth boundary and label, and each target corresponds to an Extensible Markup Language (XML) file, which is consistent with the PASCAL VOC data set. The dataset was divided into three parts: training set, validation set, and test set at a 7:2:1 ratio.

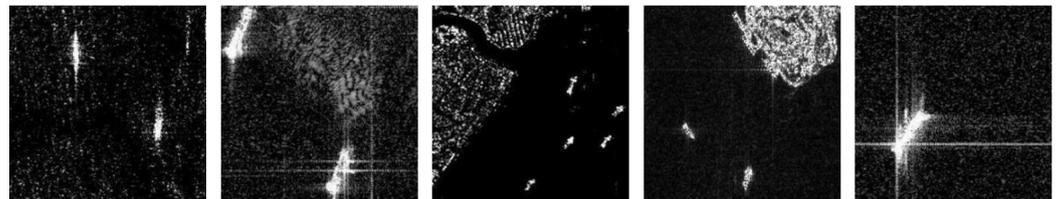


Figure 7. Sample images from the SAR ship dataset.

4.2. Experimental Results

4.2.1. Implementation Details

All of the experiments were implemented under the Darknet deep learning framework [16]. When considering the GPU capacity and training efficiency of Jeston TX1, we used a PC with a 11G memory NVIDIA GeForce GTX 1080Ti (16 GB RAM, Intel (R) Xeon(R) CPU, e3-1226 v3, @3.30 GHz) to train the model. The operating system was 64-bit Ubuntu16.04, the CUDA8.0 and its matching cuDNN were installed in the system. Subsequently we ported the well-trained model into Jeston TX1.

4.2.2. Comparison with Other Models

In order to evaluate the performance of the model, we used mAP (mean average precision) and FPS (frames per second) as the evaluation indicators for ship detection. The mAP is the average of the average accuracy of detecting each class. Because there are only ships in our experiment (i.e., only one class), the mAP was equal to the average accuracy (AP). FPS represents the number of images that can be successfully processed in one second.

We compared the performance of YOLOv3, YOLOv4, YOLOv3-tiny, YOLOv4-tiny, SSD, and Faster R-CNN by assessing their performance in mAP and FPS. The results are shown in Table 2. It was apparent that the performance of SSD and Faster R-CNN was poorer than YOLO, and YOLOv4-tiny achieved the best performance with regards to mAP. The mAP of YOLOv4 and YOLOv3 were higher than YOLOv4-tiny and YOLOv3-tiny, respectively. The YOLOv3-tiny's FPS performance was the best. The network structures of YOLOv4 and YOLOv3 are complex and have many parameters, which is suitable for more complex tasks and a larger variety of detection tasks. However, because the goal of this paper was to detect only one class of target, i.e., ships, the YOLOv4, and YOLOv3 networks were not necessary. This was especially true when considering that these models are larger and require more powerful computing platforms. The embedded computing devices deployed on spaceborne SAR satellites must be lightweight and, therefore, less powerful. Therefore, YOLOv3-tiny and YOLOv4-tiny were better suited for our on-board detection mission, and, of the two, YOLOv4-tiny was selected because it had a higher mAP.

Table 2. Comparison of different methods in FPS , mAP, and model size.

Model	FPS	mAP(%)	Model Size (MB)
YOLOv3	64	88.82	234.9
YOLOv4	55	93.21	244.2
YOLOv3-tiny	348	91.16	33.1
YOLOv4-tiny	339	93.46	22.4
SSD	76	75.46	93.2
Faster R-CNN	17	88.75	527.8

4.2.3. Results of Ship Detection with Simulated SAR Image

The method that was designed in this paper was centered on detecting ships in large-size SAR images, so we tested the robustness of the proposed method on large-size SAR images. We used simulated SAR images to evaluate the robustness of the model. The image data source was Sentinel-1, and its format was a simulated HISEA-1 SAR image format. During the simulation process, regular black bands are inevitably generated in the images because of deviations in geometric corrections, but these did not affect the actual detection results. Table 3 presents the image information.

Table 3. Simulated SAR image specific information. ¹ Location refers to the longitude and latitude of the center point of the image. The width is the number of range pixels and the height is the number of azimuth pixels.

Location ¹	Resolution	Width ¹	Height ¹	Polarization
31°1'29.21"N 122°41'36.44"E	15 m	9042	12,930	VV

In the CFAR detection, we set the false alarm rate to 0.001, and used the sliding window to extract targets that may be ships from the images. The detection result of the CFAR algorithm was binarized, and we mapped it to the original image and stored it as a 256×256 pixel image patch. Figure 8 shows examples of the output results.

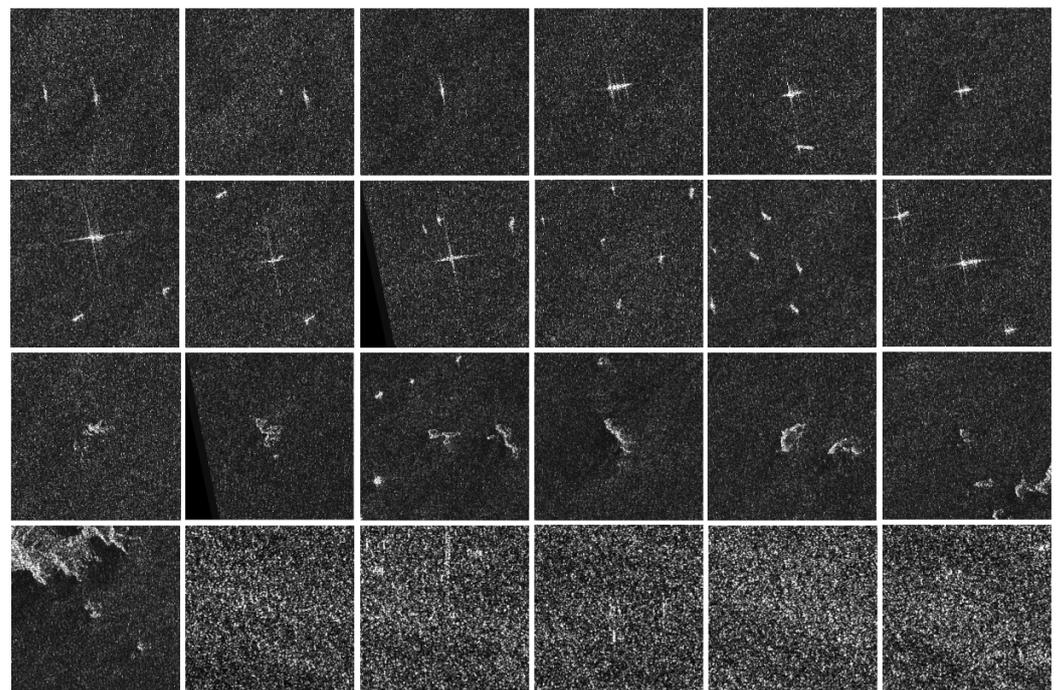


Figure 8. Examples of image patches after CFAR detection.

On this test image, the total of 673 image patches were output by the CFAR algorithm, which included image patches with real ship targets and some false alarms. From the results, it can be seen that the CFAR algorithm was able to detect most ship targets, but false alarms were prone to occur in areas with complex sea conditions, and some sea waves were commonly detected as ship targets. At the same time, because some small islands and reefs in the ocean are not connected to land, they are not segmented during pre-processing, and they can also be detected as ship targets. Secondly, there are some ship-like targets that do not fully conform to a ship's characteristics, but the CFAR detector cannot completely filter them. Additionally, there are also some very small ships that were not detected. Because

of the various false alarm sources, it is necessary to use deep learning to further assess the initial positive detection results and improve the ship detection accuracy.

A true positive (TP) indicates that the predicted result was a ship, and it actually is a ship. A true negative (TN) means that the predicted result was not a ship and that the actual result is not a ship. In our experiment, we did not add negative samples into the dataset, so the occurrence of TN in the experiment was zero. The false positive (FP) indicates that, while the predicted result was a ship, in fact there is no ship in the image. The false negative (FN) indicates that the model predicted that there was no ship in the image, but the image does contain a ship.

We used false alarm (FA), missing alarm (MA), recall, and overall accuracy (OA) to evaluate the detection performance. The definitions of these indicators are as follows:

$$FA = \frac{FP}{TP + FP} \quad (15)$$

$$MA = \frac{FN}{TP + FN} \quad (16)$$

Precision is the ratio of true positives to the identified image:

$$precision = \frac{TP}{TP + FP} \quad (17)$$

Recall represents the ratio of the number of detected ship targets (TP) to all ship targets in the image.

$$Recall = \frac{TP}{TP + FN} \quad (18)$$

The overall accuracy is the most intuitive performance measure and it is a ratio of correctly predicted observations to the total observations.

$$OA = \frac{TP + TN}{TP + TN + FP + FN} \quad (19)$$

Figure 9 and Table 4 shows the detection results of our proposed method and only using YOLOv4-tiny. The results verified that our proposed method was correct and effective in terms of accuracy and efficiency. It can be seen that the ship detection system we proposed was robust, with a detection accuracy rate of 88.58%, a false alarm rate of 14.12%, and a missed detection rate of 11.42%, some false alarms and missing ships are shown in Figure 10. Furthermore, the running time was improved and it met the mission requirements for real-time ship detection using the on-board platform of the satellite.

Table 4. The ship detection results of the the simulated image.

Method	Detected Ships	TP	FP	FN	Precision	Recall	OA	FA	MA	Running Time (s)
Proposed method	262	225	37	29	85.9%	88.6%	77.3%	14.1%	11.4%	67.18s
Only YOLOv4-tiny	274	232	42	22	84.7%	91.3%	78.4%	15.3%	8.7%	156.83s

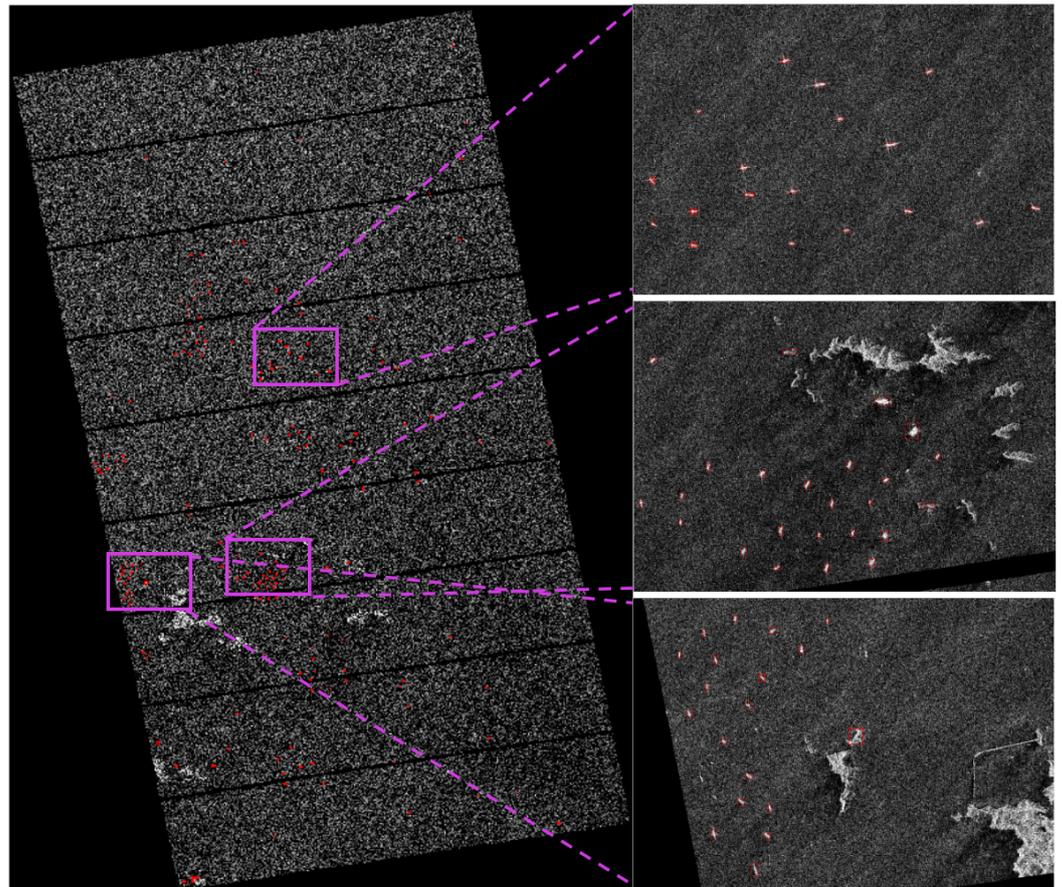


Figure 9. Ship detection results of the simulated SAR image.

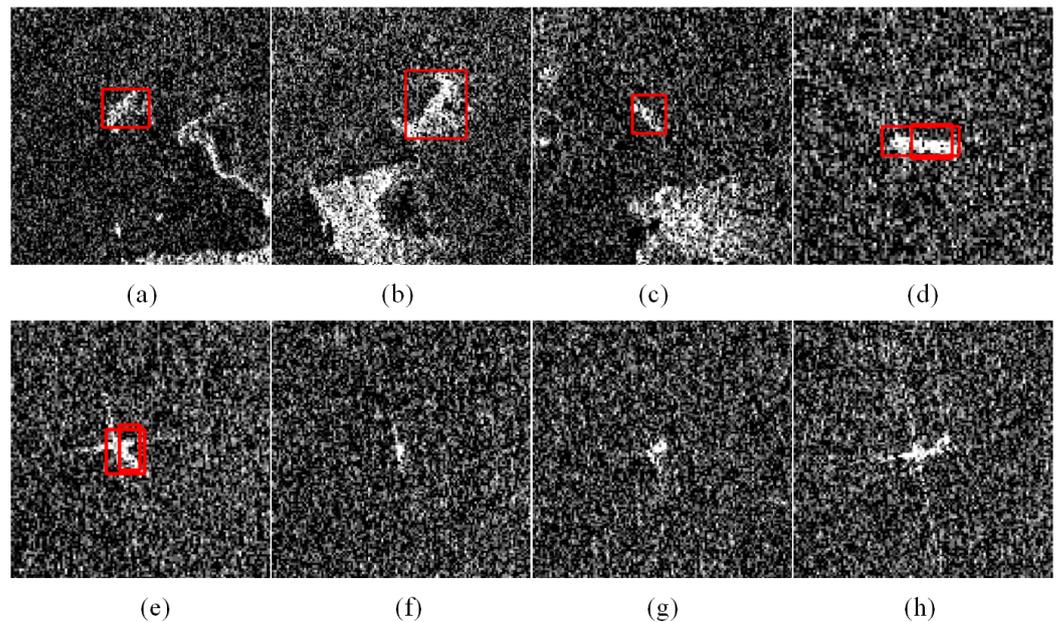


Figure 10. Examples of false alarms (a–e) and missing ships (f–h).

4.2.4. Results of Ship Detection with HISEA-1 Image

With the successful operation of HISEA-1 in orbit, we have obtained some actual results of on-board ship detection. We selected a relatively representative image for

verification purposes due to the limited data. Table 5 shows the information for the HISEA-1 test image. Figure 11 shows the test results.

Table 5. The information of the HISEA-1 image.

Acquisition Time	Resolution	Width	Height	Polarization
2020.12.25	3 m	9980	6607	VV

There were relatively few ship targets in this test image, and it was obvious that our method was able to detect all of the ships on the sea, as shown in Figure 11. However, near the harbor, the model triggered a false alarm, which may have been caused by the building having a similar backscattering signature to the ships.

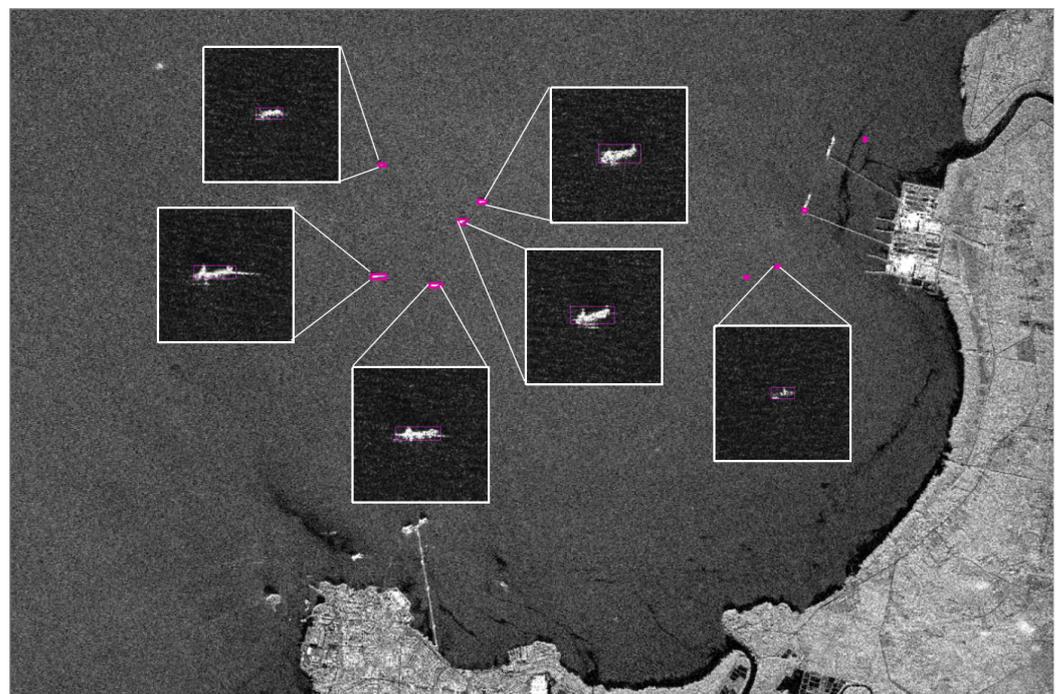


Figure 11. The ship detection results on HISEA-1 image. The purple boxes are the detection results of our method and the white boxes are the target patches extracted by our network.

5. Discussion

The above experiments verified the validity of our scheme. We can deploy our network scheme to the Jetson TX1 on the lightweight SAR satellite for on-board ship detection. The combination of the classic CFAR algorithm and YOLOv4-tiny lightweight deep learning network greatly improved the speed of ship detection. After extracting the ship targets, the coordinates of the upper left and lower right corners of the images are collected and latitude and longitude coordinates of the ship are obtained through coordinate conversion. Only the target image patches and geographic coordinates will be transmitted to the ground receivers. This on-board ship detection method greatly reduces the amount of data that must be transferred to the ground receiver, which helps to reduce the pressure on the ground receiver. However, if required for special missions, users on the ground can control the satellite and obtain images of a region of interest, and the useful information will be transmitted to the ground after on-board processing. With these data, ground users will be able to make better and faster decisions in emergency situations.

In addition, as shown above, our method achieved a rapid and accurate ship detection in SAR images. In order to obtain better ship detection results, subsequent work will need to label the ship chips of original images that were downloaded from the satellite at an early

stage and train the model. Finally, the model can be updated or a new model can be uploaded to the satellite to achieve more accurate detection results. Furthermore, in addition to the role of the deep learning model in determining detection accuracy, the CFAR algorithm also impacts the detection accuracy. Therefore, our future work will continue to improve the CFAR algorithm.

6. Conclusions

In this paper, we proposed an on-board real-time ship detection scheme based on the CFAR algorithm and lightweight deep learning for HISEA-1 SAR imagery, and constructed a ground system to verify the feasibility of our scheme. The scheme uses CFAR to perform initial ship detection followed by the YOLOv4-tiny method to obtain the final results. Subsequently, the entire algorithm was ported into the Jetson TX1 GPU and, through a reasonable hardware acceleration strategy, our scheme achieved high accuracy and efficiency on the ground verification platform. The experiments showed that this scheme demonstrated good applicability for real-time on-board ship detection on lightweight SAR satellites. In our future work, we will continue to improve the CFAR algorithm while also improving the YOLOv4-tiny network to further maximize the computational efficiency. When more data from the HISEA-1 are obtained, we will focus on labelling the ship chips to improve the accuracy of target detection.

Author Contributions: P.X., B.Z., F.W. and R.Z. conceived of this study. P.X. performed the experiments and wrote the manuscript. B.Z., F.W. and R.Z. supervised the work and revised the manuscript. Q.L., K.Z., X.D. and C.Y. helped to acquire the data of experiments, and contributed with field experience in HISEA-1 satellite and on-board processing. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by Academy for Multidisciplinary Studies, Capital Normal University and the National Natural Science Foundation of China under Grant 42071444.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Yao, Y.; Jiang, Z.; Zhang, H.; Zhou, Y. On-Board Ship Detection in Micro-Nano Satellite Based on Deep Learning and COTS Component. *Remote Sens.* **2019**, *11*, 762. [[CrossRef](#)]
2. Cui, Z.; Dang, S.; Cao, Z.; Wang, S.; Liu, N. SAR Target Recognition in Large Scene Images via Region-Based Convolutional Neural Networks. *Remote Sens.* **2018**, *10*, 776. [[CrossRef](#)]
3. Chang, Y.L.; Anagaw, A.; Chang, L.; Wang, Y.; Hsiao, C.Y.; Lee, W.H. Ship detection based on YOLOv2 for SAR imagery. *Remote Sens.* **2019**, *11*, 786. [[CrossRef](#)]
4. Li, D. Towards Geo-spatial Information Science in Big Data Era. *Acta Geod. Cartogr. Sin.* **2016**, *45*, 379–384.
5. Li, D.; Shen, X.; Gong, J.; Zhang, J.; Lu, J. On Construction of China's Space Information Network. *Geomat. Inf. Sci. Wuhan Univ.* **2015**, *40*, 711–715.
6. Zhang, B. Intelligent remote sensing satellite system. *J. Remote Sens.* **2011**, *15*, 415–431.
7. Davis, C.O.; Horan, D.M. On-orbit calibration of the Naval EarthMap Observer (NEMO) coastal ocean imaging spectrometer (COIS). In Proceedings of the International Symposium on Optical Science and Technology, San Diego, CA, USA, 30 July–4 August 2000.
8. Wang, M. Stream-computing Based High Accuracy On-board Real-time Cloud Detection for High Resolution Optical Satellite Imagery. *Acta Geod. Cartogr. Sin.* **2018**, *47*, 760.
9. Gao, G.; Liu, L.; Zhao, L.; Shi, G.; Kuang, G. An Adaptive and Fast CFAR Algorithm Based on Automatic Censoring for Target Detection in High-Resolution SAR Images. *IEEE Trans. Geosci. Remote Sens.* **2009**, *47*, 1685–1697. [[CrossRef](#)]
10. Huang, X.; Yang, W.; Zhang, H.; Xia, G. Automatic Ship Detection in SAR Images Using Multi-Scale Heterogeneities and an a Contrario Decision. *Remote Sens.* **2015**, *7*, 7695–7711. [[CrossRef](#)]
11. Ji, Y.; Jie, Z.; Junmin, M.; Xi, Z. A new CFAR ship target detection method in SAR imagery. *Acta Oceanol. Sin.* **2010**, *29*, 12–16. [[CrossRef](#)]

12. Ai, J.; Qi, X.; Yu, W.; Deng, Y.; Liu, F.; Shi, L. A New CFAR Ship Detection Algorithm Based on 2-D Joint Log-Normal Distribution in SAR Images. *IEEE Geosci. Remote Sens. Lett.* **2010**, *7*, 806–810. [[CrossRef](#)]
13. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Region-Based Convolutional Networks for Accurate Object Detection and Segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *38*, 142–158. [[CrossRef](#)]
14. He, K.; Zhang, X.; Ren, S.; Sun, J. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2014**, *37*, 1904–1916. [[CrossRef](#)]
15. Girshick, R. Fast R-CNN. Computer Science. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015.
16. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1137–1149. [[CrossRef](#)]
17. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016.
18. Redmon, J.; Farhadi, A.B.T.I.C.o.C.V.P.R. YOLO9000: Better, Faster, Stronger. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 6517–6525.
19. Redmon, J.; Farhadi, A. YOLOv3: An Incremental Improvement. *arXiv* **2018**, arXiv:1804.02767.
20. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S. SSD: Single Shot MultiBox Detector. In Proceedings of the 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016.
21. Wang, Y.; Wang, C.; Zhang, H.; Dong, Y.; Wei, S. Automatic Ship Detection Based on RetinaNet Using Multi-Resolution Gaofen-3 Imagery. *Remote Sens.* **2019**, *11*, 531. [[CrossRef](#)]
22. Kang, M.; Leng, X.; Lin, Z.; Ji, K.B.T.I.W.o.R.S.w.I.P.R. A modified faster R-CNN based on CFAR algorithm for SAR ship detection. In Proceedings of the 2017 International Workshop on Remote Sensing with Intelligent Processing (RSIP), Shanghai, China, 18–21 May 2017.
23. Li, H.; Wang, C.; Zhang, H.; Wu, F. Automatic SAR Sea-land Segmentation Based on Sea Chart Information. *Remote Sens. Technol. Appl.* **2009**, *6*, 731–736.
24. Foley, J.; van Dam, A.; Feiner, S.; Hughes, J. *Computer Graphics: Principles and Practice in C*; Wiley: New York, NY, USA, 1995; Volume 702.
25. Han, J.; Duan, X.; Chang, Z. Target Segmentation Algorithm Based on SLIC and Region Growing. *Comput. Eng. Appl.* **2021**, *57*, 213–218.
26. Lu, T.; Zhang, J.; Ji, Y.; Zhang, X.; Meng, J. Ship Target Detection Algorithm Based on G^0 Distribution for SAR Images under Rough Sea Conditions. *Adv. Mar. Sci.* **2011**, *29*, 186–195.
27. Ziou, D. Automatic Detection for Ship Targets in RADARSATSAR Images from Coastal Regions. In Proceedings of the Vision Interface '99, Trois-Rivieres, QC, Canada, 18–21 May 1999.
28. Su, S.; Chen, H. Parameter Estimation on Sea Clutter Compound K-distribution Model. *Comput. Appl. Softw.* **2014**, *31*, 273–275.
29. Bochkovskiy, A.; Wang, C.Y.; Liao, H.y. YOLOv4: Optimal Speed and Accuracy of Object Detection. *arXiv* **2020**, arXiv:2004.10934.
30. Wang, C.Y.; Liao, H.Y.M.; Wu, Y.H.; Chen, P.Y.; Hsieh, J.W.; Yeh, I.-H. CSPNet: A New Backbone that can Enhance Learning Capability of CNN. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, Seattle, WA, USA, 14–19 June 2020.
31. Wang, Y.; Wang, C.; Zhang, H.; Dong, Y.; Wei, S. A SAR Dataset of Ship Detection for Deep Learning under Complex Backgrounds. *Remote Sens.* **2019**, *11*, 765. [[CrossRef](#)]