A Deep Detection Network Based on Interaction of Instance Segmentation and Object Detection for SAR Images

Zitong Wu, Biao Hou, Bo Ren, Zhongle Ren, Shuang Wang and Licheng Jiao

The Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education, School of Artificial Intelligence, Xidian University, Xi’an 710071, China; wuzitong@stu.xidian.edu.cn (Z.W.); boren@xidian.edu.cn (B.R.); zlren@stu.xidian.edu.cn (Z.R.); shwang@mail.xidian.edu.cn (S.W.); lchjiao@mail.xidian.edu.cn (L.J.)
* Correspondence: houbiao@mail.xidian.edu.cn

Abstract: Ship detection is a challenging task for synthetic aperture radar (SAR) images. Ships have arbitrary directionality and multiple scales in SAR images. Furthermore, there is a lot of clutter near the ships. Traditional detection algorithms are not robust to these situations and easily cause redundancy in the detection area. With the continuous improvement in resolution, the traditional algorithms cannot achieve high-precision ship detection in SAR images. An increasing number of deep learning algorithms have been applied to SAR ship detection. In this study, a new ship detection network, known as the instance segmentation assisted ship detection network (ISASDNet), is presented. ISASDNet is a two-stage detection network with two branches. A branch is called an object branch and can extract object-level information to obtain positioning bounding boxes and classification results. Another branch called the pixel branch can be utilized for instance segmentation. In the pixel branch, the designed global relational inference layer maps the features to interaction space to learn the relationship between ship and background. The global reasoning module (GRM) based on global relational inference layers can better extract the instance segmentation results of ships. A mask assisted ship detection module (MASDM) is behind the two branches. The MASDM can improve detection results by interacting with the outputs of the two branches. In addition, a strategy is designed to extract the mask of SAR ships, which enables ISASDNet to perform object detection training and instance segmentation training at the same time. Experiments carried out on two different datasets demonstrated the superiority of ISASDNet over other networks.

Keywords: synthetic aperture radar (SAR); object detection; convolutional neural network (CNN); instance segmentation

1. Introduction

Synthetic aperture radar (SAR) is an active microwave imaging equipment that can work all day and in all weather [1]. As the performance of the SAR system gradually improves, more and more high-resolution and high-quality SAR images can be acquired. Many countries have developed their own SAR systems, such as TerraSAR-X, COSMOS-SkyMed, RADARSAT-2, ALOS-PALSAR, Sentinel-1, and Gaofen-3 [2]. The application value of SAR is increasing in various fields. Ship detection is very meaningful, as it can provide basic information for ship traffic management [3,4], the fishing industry [5,6], and safe navigation [7–9]. The SAR system can continuously observe the sea area for a long time without the interference from clouds, fog, rainfall, or snowfall. Therefore, SAR ship detection has attracted the attention of researchers in various countries.

There is a lot of noise in SAR images, which affects ship detection. Ships parked in the port are affected by land clutter, which makes ship detection more difficult. Moreover, small ships are easy to ignore, while dense ships always appear as a bright spot in SAR images, which makes it difficult to identify individual ships. In most cases, the backscattered signal of ships is much larger than that on the sea surface. Ships are brighter than the surrounding
area in the SAR image. Thus, ship detection can be regarded as searching the pixel area where the intensity level of the scattering signal is greater than the given threshold. The constant false alarm rate (CFAR) algorithm [10] is one of the most famous intensity threshold-based algorithms. According to the background signal, the CFAR adjusts the decision threshold adaptively to ensure that the recognition result has a constant false alarm probability. This kind of threshold-based algorithm is not robust against complex scenes in high-resolution SAR images. With the great achievement of deep learning, researchers have designed many ship detection algorithms based on the convolutional neural network (CNN) [11–13]. Object detection algorithms based on the CNN are generally divided into two categories: one-stage algorithms and two-stage algorithms. You only look once (YOLO) [14] and the single-shot multi-box detector (SSD) [15] are classical one-stage algorithms. Two-stage algorithms include regional CNN (R-CNN) [16] and fast R-CNN [17]. Generally speaking, the accuracy of two-stage algorithms is higher, but their calculation cost is also higher [18]. On the contrary, the one-stage algorithm has faster speed and adopts end-to-end training.

Object detection and instance segmentation are two basic tasks in computer vision [19]. They are closely related, i.e., one is object-level detection and the other is pixel-level extraction. Most previous works have not fully explored the relationship between them. Understanding the interaction between object-level information and pixel-level information is significant to improve network performance. In ship detection, the most important task is to locate an inclusive and tight bounding box for the ship. Localization errors are easily produced in ship detection, as shown in Figure 1. In Figure 1a, the box does not contain the ship completely. In Figure 1b, the box does not hold the ship tightly. These errors come from the fact that the network does not know the whole object in regression. However, pixel-level information can be used to reduce these localization errors. Ideally, the smallest bounding rectangle of the mask is the bounding box of the object. Instance segmentation is a combination of object detection and semantic segmentation, and its purpose is to predict the categories of all pixels in each object region. The existing instance segmentation algorithms are basically two-stage algorithms, such as Mask R-CNN [20]. These instance segmentation algorithms have two independent branches for object detection and instance segmentation. However, they cannot make object-level information interact with pixel-level information to improve network performance.

![Figure 1](image1.png)

**Figure 1.** Localization errors in SAR ship detection. (a) The box does not contain the ship completely. (b) The box does not hold the ship tightly.

In the existing open SAR ship datasets, only the vertical bounding box is used as the label because the backscattered signal of a ship is much larger than that on the sea surface. With the help of label information, the ship’s pixels can be easily separated from the background through some image processing. Although a mask obtained in this way is not very accurate, it can roughly locate the outline of the ship. Hence, ship detection and ship instance segmentation can be trained simultaneously. In this work, an instance segmentation assisted ship detection network (ISASDNet) is presented. ISASDNet is based on the Mask R-CNN and is a two-stage detection network that has two branches. Pixel-level
information can be extracted to promote ship detection in ISASDNet. The contributions of this work are summarized below.

1. A strategy to extract the mask is designed. The approximate ship contour is obtained by a threshold-based segmentation method. Through this strategy, the mask does not exceed the bounding box, which is the labels of the dataset. Ship detection and ship instance segmentation can thus be carried out synchronously when training the network.

2. A global reasoning module is designed to improve the accuracy of predicting the ship mask. Features are mapped to the interaction space. The relationship between ship and background can be regarded as a two-node graph. The global relationship can be obtained through this module.

3. A module that uses the results of instance segmentation to enhance object detection is designed. The positioning coordinate can be regarded as a classification task. After obtaining rough target recognition results and instance segmentation results, the posterior probability of coordinate points is calculated to obtain the final result. This module can mine the relationship between pixel-level information and object-level information.

The rest of this paper is arranged as follows. Some related work is introduced in Section 2. The proposed network is described in Section 3. Then, the proposed network is compared with other algorithms in Section 4. The results of some analytical experiments are also presented in Section 4 Finally, Section 5 concludes the paper.

2. Related Work

2.1. Traditional Ship Detection Methods

Due to the different radar reflection characteristics between the ship and sea water, in SAR images, the ship is bright while the water is dark. However, the contrast between the ships and the sea background changes constantly, and this requires the ship detection algorithm to have adaptability and maintain a constant false alarm rate. The CFAR algorithm is based on a statistical model, is widely employed by researchers, and has the characteristics of fast speed, adaptive threshold, simple design, etc. Wang et al. used the internal Hermitian product to obtain a new detector [21]. The detector can apply a threshold to obtain SAR ship discrimination. A data-driven Parzen window kernel function was utilized to approximate the histograms of SAR images in [22]. SAR ship objects have also been filtered by the given threshold of the CFAR. An et al. proposed a modified iterative truncation algorithm for the CFAR [23]. This method searched target pixels and their four-connected neighborhood pixels to estimate local sea clutter distributions. Li et al. used weighted information entropy to describe the statistical characteristics of super pixels [24]. They separated the ship target from the background super pixels by changing the threshold value. A novel decomposition approach was presented to analyze scattering between ships and sea background in [25]. This decomposition approach has been combined with CFAR to detect ships. Lang et al. extracted the representation of pixels by using spatial and intensity information, and the separability of ship and background was significantly improved [26]. In addition to statistical methods, the traditional SAR ship detection includes multi-scale-based methods [27], template matching methods [28], full polarization-based methods [29,30], etc. Pastina et al. proposed a processing chain based on the cell average and generalized likelihood ratio test for the detection of ships [27]. Ouchi et al. proposed that the multilook images of ships have higher coherence than that of surrounding sea surface [31]. They used a small moving window to calculate the cross-correlation value between the two images, which can extract the ships. Tello et al. proposed an approach for ship detection based on the analysis of SAR images by discrete wavelet transform [32]. Although these traditional methods have achieved good results, their robustness decreases as SAR image resolution increases.

2.2. Object Detection Using CNNs

With the development of deep learning, CNNs have shown strong performance in object detection. All kinds of detectors based on the CNN are composed of two parts:
a backbone and a head [33]. The backbone is usually the VGG [34], ResNet [35], or DenseNet [36]. The head is classified into two types: the one-stage algorithm, and the two-stage algorithm. The R-CNN [16], fast R-CNN [17], and faster R-CNN [37] are the most typical two-stage algorithms. Two-stage algorithms produce region proposals that may contain objects, and then classify and calibrate the region proposal to produce the final detection results. The most representative one-stage algorithms are the YOLO [14] series, SSD [15], and RetinaNet [38]. Unlike two-stage algorithms, the one-stage algorithms do not generate proposal boxes. In addition, many embedded network modules have been designed. For example, Lin et al. exploited the inherent multi-scale convolutional network, called a feature pyramid network (FPN), to improve the accuracy of object detection [39]. Furthermore, a path aggregation network can boost the quality of prediction by accelerating information flow and integrating the features of different levels [40].

2.3. Ship Detection for Optical Remote Sensing Image

The imaging mechanism of optical image is different from that of SAR image. Optical image is the image data obtained by visible light sensors. Optical image usually contains gray information of multiple bands. It includes a lot of color information, shape information, texture information, etc. Researchers have designed a large number of algorithms for ship detection in optical remote sensing images. A network that searched the head of a ship globally and generated smaller proposal boxes was proposed for inshore ship detection [41]. Yan et al. presented a data enhancement strategy using simulated ship images to augment the positive training samples [42]. Additionally, this strategy improved the training accuracy of faster R-CNN. For ship rotation detection, a dual branch regression network was designed, which can extract features with different aspect ratios and integrate multi-scale features [43]. Liu et al. designed a multiregion feature-fusion module to improve faster R-CNN, and used multitask learning to classify, locate, and regress ships [44]. The network proposed by Ma et al. can generate rotating region suggestions by constructing central region prediction and orientation classification [45]. Feng et al. proposed a ship detection and classification framework by introducing a new sequence local context module [46].

2.4. SAR Ship Detection with Deep Learning

SAR can observe the earth all day, unlike optical sensors that cannot work at night. Ships in SAR images are different from those in optical images. SAR image lacks color information, texture information, shape information, etc. Additionally, there is a lot of noise in SAR image. It is difficult for researchers without relevant knowledge to label SAR images. This results in the scarcity of labeled SAR ship data. Thus, ship detection in SAR images is more challenging. Many deep learning algorithms have been applied to ship detection in SAR images. Fan et al. embedded a multi-level features extractor into the Faster R-CNN for polarimetric SAR ship detection [47]. A dense attention pyramid network that densely connected the attention convolutional module to each feature map was presented for SAR ship detection [18]. Meanwhile, a fully convolutional network was designed for pixel-wise ship detection in polarimetric SAR images [48]. A spatial attention block and a split convolution block were embedded in the feature pyramid network [49]. The feature pyramid network can accurately detect ship objects against a complex background. Wei et al. designed a high-resolution feature pyramid network that connected high-to-low resolution features for ship detection [50]. A multi-scale adaptive recalibration network was proposed to solve the problem of ships with different sizes and dense berthing [51]. Hou et al. proposed a one-stage SAR object detection method to address the low confidence of candidates and false positives [52]. Kang et al. proposed an algorithm combining CFAR with faster R-CNN [53]. This method used the object proposals generated by faster R-CNN as the protection window of CFAR to extract small objects. Zou et al. designed a generative adversarial network with multi-scale loss term and combined it with YOLOv3 to improve the accuracy of SAR ship detection [54].
3. Methodology

In this section, the two-branch instance segmentation assisted ship detection network (ISASDNet) is firstly introduced. In the existing open datasets of SAR ship detection, only the bounding boxes are used as labels. In order to train ISASDNet, we designed a strategy that extracts the mask of ships. Then, we describe two modules that can improve the performance of ISASDNet.

3.1. Architecture

ISASDNet can perform object detection and instance segmentation simultaneously. It has two branches, as shown in Figure 2. The backbone network of ISASDNet is a combination of the ResNet, FPN, and region proposal network (RPN) \[37\]. The ResNet and FPN can extract multi-scale features of a SAR image. Meanwhile, RPN can generate the region proposal for the ships. The object branch focuses on locations and object categories and is composed of full connection layers and ROIAlign layers that fuse multi-scale features. The pixel branch is a global reasoning module (GRM) focused on pixel-level information to predict masks of ships. Our network is expected to learn a graph like global relationship between the ship region and background region, so as to represent the large contrast between ships and the sea surface. The GRM can map features to interaction space to learn the interaction between the ship and background. In interaction space, the relationship between the ship and background is regarded as a two-node graph. The adjacency metrics learned from graph can well describe the relationship. The output of the object branch is a rough object detection result, while the pixel branch outputs the instance segmentation results of ships. The module behind the two branches of ISASDNet is called the mask assisted ship detection module (MASDM). In the MASDM, the target localization can be regarded as a classification task. The outputs from the two branches are fed into the MASDM. Finally, according to Bayes theorem, the MASDM can adjust the bounding boxes to obtain the final results.

![Figure 2. Architecture of ISASDNet, which has a two-branch structure, i.e., an object branch and a pixel branch. The backbone network is a combination of ResNet, FPN, and RPN. The object branch can obtain rough object detection results. The pixel branch is based on global relational inference and can obtain the instance segmentation results. MASDM can fuse the results from these two branches to obtain more accurate object detection results.](image)

3.2. Mask Extraction Strategy

Vertical bounding boxes are usually available as labels instead of masks in SAR ship detection datasets. In fact, the pixels of SAR ships are quite different from the background. Through some well-designed strategies, it is easy to extract the ship’s contour as the mask...
with the help of label information from the datasets. The mask extraction strategy consists of four steps (described below), and the processing rendering is shown in Figure 3.

![Figure 3. Processing rendering of mask extraction strategy; the green box is the label. (a) Original image. (b) Image slice. (c) Binary image after thresholding. (d) Binary image after morphology processing. (e) Final mask.](image)

3.2.1. Image Slices

Figure 3a is the original image, and the green bounding box is the label. The length and width of the given labels are each expanded by 50% to cut original image. The sliced image containing the object is as shown in Figure 3b. The area beyond the sliced image is filled with padding. In this way, the background around the ships in the original image can be preserved more, which is convenient for subsequent threshold segmentation, corrosion, and expansion operations.

3.2.2. Thresholding

We use an adaptive threshold segmentation method to generate binary images. Even if there are all kinds of clutter in a SAR image, the pixel value of the ship is larger than that of the background. Sort all pixels by pixel value in sliced image. We assume that $\beta$ of the pixels with the highest pixel value are ships, and the rest comprise the background. $\beta$ is a hyperparameter. Therefore, the threshold $T_1$ is defined as follows:

$$ T_1 = \min_t \left[ \sum_{p \in P} \text{sgn}(p - t) \leq \beta \cdot I(P) \right], $$

where $\text{sgn}(\cdot)$ is a signed function. In the experimental part, we analyzed the data of the training set. When $\beta$ was between 0.35 and 0.4, the ship pixels can be extracted completely. In order to ensure the integrity of the ships, $\beta$ was determined to be 0.4. In other words, 40% of the pixels with the highest pixel value were regarded as ships. Ridler et al. proposed the IsoData threshold segmentation algorithm [55], which is a classical clustering method. It uses the variance within and between clusters to inform further clustering. When the number of samples in a cluster is too small or the distance between two clusters is too close, the clusters are merged; meanwhile, when the inner variance of a cluster is too large, the cluster is split. In the IsoData threshold segmentation algorithm, a random threshold is given to segment the image into objects and background firstly. Then, the mean value of these two parts is calculated and iterated until the threshold is greater than the composite mean value. The threshold $T_2$ can be calculated quickly by `skimage.threshold_isodata` in Python [56]. The final threshold $T$ is:

$$ T = \begin{cases} T_1, & T_1 > T_2 \\ T_2, & T_1 \leq T_2 \end{cases} $$

The binary image (Figure 3c) is obtained according to $T$. 
3.2.3. Morphological Processing

In the binary image, there are still a lot of clutter and noise in the target contour area. These noises and clutter can be removed by twice average filtering. Then, corrosion and expansion can make the mask more complete (see Figure 3d).

3.2.4. Output Mask

Since it is still possible for the mask to go beyond the label box, only the mask in the bounding box is reserved. The final mask is shown in Figure 3e.

3.3. Global Reasoning Module

Relational reasoning between distant regions of arbitrary shape is crucial for object detection [57]. The CNN has shown extraordinary ability in many computer vision tasks and is good at calculating local relations. However, it needs to stack multiple convolutions to capture the global relations between remote areas. Humans can easily understand the relationship between different regions of an image. The structure of a graph can better describe the relationship between different regions than a figure. The graph convolution network can project the region of interest into an interaction space to infer the global relationship [58]. This reasoning process is similar to human cognition. Chen et al. proposed a graph-based unit that can transform features in coordinate space and interaction space for global reasoning [57]. This unit combines a graph convolution with an ordinary convolution and can be easily embedded in various networks. In SAR images, ships are sometimes very small, and too many convolution layers may reduce the accuracy and speed of the inference process. The task of SAR ship instance segmentation can be regarded as two-class segmentation. The relationship between the ship and background can be regarded as a two-node graph. The graph is formed by projecting features from coordinate space to interaction space, which can better infer the global relationship between ship and background. The features in the interaction space are mapped back to the coordinate space by back projection. This global relationship is conducive to image segmentation. Therefore, a global reasoning module (GRM) embedded in ISASDNet is devised to improve the accuracy of ship instance segmentation.

The global relational inference layers are based on the unit proposed in [57]. Figure 4 presents the structure of the global relational inference layer. An input feature \( F \in \mathbb{R}^{L \times C} \) is fed into a global relational inference layer, where \( L = W \times H \). \( W \) is the width of the feature, \( H \) is the height of the feature, and \( C \) is the channel of the feature. A function \( F' = \phi(F) \in \mathbb{R}^{L \times \frac{C}{2}} \) is used to reduce the input dimension to half of the original dimension. To improve the calculation speed and the capacity of the projection function, \( \phi(\cdot) \) can be done by convolution. Similarly, a projection weight \( B \) can be obtained by convolution:

\[
B = \theta(F) \in \mathbb{R}^{2 \times L}
\]

A new projection feature \( V \) is obtained in interaction space:

\[
V = B \cdot F' \in \mathbb{R}^{N \times \frac{C}{2}}
\]

Here, \( N = 2 \) is the number of nodes that represent the ship and the background. The nodes need to interact to learn the relationship between the ship and the background. \( G \) and \( A \) represent the \( N \times N \) node adjacency matrix, and \( U \) represents the state update function. The graph interaction is defined as follows:

\[
Z = GVU = (I - A)V \cdot U
\]
This interaction between the nodes can be completed by two 1D convolution layers along channel-wise and node-wise directions. Then, node-feature \( Z \in \mathbb{R}^{N \times C} \) needs to be transformed into feature \( W' \in \mathbb{R}^{L \times \frac{C}{2}} \) in coordinate space. The reverse projection matrix \( D \) can be regarded as the transposition of \( B \). Similar to the transformation from coordinate space to interaction space, the transformation from interaction space to coordinate space can also be completed by convolution operation. \( W' \) adds dimension to obtain feature \( W \in \mathbb{R}^{L \times C} \).

**Figure 4.** Global relational inference layer. Two convolutions are for dimension reduction and expansion. Projection matrix \( B \) and back projection matrix \( D \) transpose each other. They can be obtained by a convolution operation. The graph interaction has two 1D convolutions, one along the channel direction and the other along the node direction.

The proposed GRM is shown in Figure 5. It is composed of an ROIAlign layer, a convolution layer, four global relational inference layers, and four deconvolution layers. The ROIAlign layer can fuse multi-scale features from the backbone network. Meanwhile, the global relational inference layers can capture global relationships between the ship and the background. Lastly, the four deconvolution layers transform features into the same size as the original image.

**Figure 5.** Structure of the GRM. It is composed of a convolution layer, four graph convolution layers, four deconvolution layers, and an ROIAlign layer.

### 3.4. Mask Assisted Ship Detection Module

The object branch can provide a rough bounding box for the ship. By calculating the predicted mask of the pixel branch, the vertical minimum circumscribed rectangle can also be obtained as the bounding box. The MASDM is designed to improve detection results by
using the bounding boxes from these two branches. The determination of coordinates can be regarded as a task of discrete variable classification. The bounding box can be simplified to the calculation of four coordinates (left, top, right, and bottom). Here, we only analyze the abscissa \( x \) of the upper-left corner. The ordinate is calculated in the same way, and the other three coordinates are calculated similarly. The location \( x \) is the argmax of the probability of a coordinate:

\[
x = \arg \max_i P(X = i | X_O = j, X_p = k)
\]  

(7)

where \( X \) is the random variable for the coordinate of left, \( X_O = j \) means that the abscissa of the left boundary is \( j \) in the object branch, \( X_p = k \) means that the abscissa of the left boundary is \( k \) in the pixel branch, and \( P(X = i | X_O = j, X_p = k) \) denotes the posterior probability given the result of object branch \( X_O = j \) and the result of pixel branch \( X_p = k \).

According to Bayes theorem, Equation (7) can be transformed into

\[
P(X = i | X_O = j, X_p = k) = \frac{P(X = i)P(X_O = j, X_p = k | X = i)}{\sum_{t=1}^{\infty} P(X = t)P(X_O = j, X_p = k | X = t)}
\]  

(8)

where \( P(X = i) \) and \( P(X_O = j, X_p = k | X = i) \) are the prior and likelihood probabilities, respectively, and \( w \) is the width of the image.

A Gaussian distribution is used to calculate probability \( P(X = i) \):

\[
P(X = i) = \alpha e^{-(\mu - \mu)^2 / 2\delta^2}
\]  

(9)

where \( \alpha \) is the normalization coefficient. This Gaussian distribution is related to the image size and the results of two branches. Therefore,

\[
\mu = \frac{i + k}{2}, \quad \delta = \gamma \frac{w_O + w_p}{2}
\]  

(10)

where \( \gamma \) is a weight factor, \( w_O \) is the width of the proposal box from the object branch, and \( w_p \) is the width of the proposal box from pixel branch.

Assuming that \( X_O = j \) and \( X_p = k \) are independent, the likelihood probability can be defined as follows:

\[
P(X_O = j, X_p = k | X = i) = P(X_O = j | X = i) \cdot P(X_p = k | X = i)
\]  

(11)

It is hard to calculate \( P(X_O = j | X = i) \) and \( P(X_p = k | X = i) \) directly. Thus, two 1D convolution kernels are learned to calculate them. First, we flatten the predicted mask to get a vector of length \( w \). The flatten process is shown in Figure 6 and should be carried out for each ship. A 1D convolution kernel slides on the flattened vector to get the probability \( P(X_p = k | X = i) \) of each point. The bottom row of images in Figure 6 shows the flatten process of the object branch. The object branch produces detection boxes for each ship. This flatten process needs to be applied to each ship. All parts of the original image beyond detection box are set to zero. Then, each column is maximized to produce a vector of length \( w \). Another 1D convolution kernel slides on the vector to obtain \( P(X_O = j | X = i) \). The length \( m \) of the convolution kernel is a super parameter. The coordinate with the largest probability value is the final result. The four coordinates of the bounding box can be calculated in this way.
Figure 6. Flatten process in two branches. The top row is the flatten process of the mask in the pixel branch. The bottom row is the flatten process of the result in the object branch. All parts of the original image beyond box are set to zero. Then, each column is maximized to produce a vector.

3.5. Loss Function

The proposed ISASDNet is trained by the following loss function:

\[ L = L_{\text{cls}} + \lambda_1 L_{\text{box}} + \lambda_2 L_{\text{mask}} \]  

where \( \lambda_1 \) and \( \lambda_2 \) are weight coefficients. The classification loss, bounding box loss and mask loss are identical as those defined in [20]. Specifically, \( L_{\text{cls}} \) is log loss:

\[ L_{\text{cls}}(p, u) = -\log p_u \]  

where \( p \) is the predicted class and \( u \) is the true class. Suppose that \( v = (v_x, v_y, v_w, v_h) \) is the ground truth of bounding box and \( t = (t_x, t_y, t_w, t_h) \) is the predicted result. For bounding box regression, \( L_{\text{box}} \) is defined as:

\[ L_{\text{box}} = \sum_{i \in \{x,y,w,h\}} \text{smooth}_{L_1}(t_i - v_i) \]  

in which

\[ \text{smooth}_{L_1}(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise} \end{cases} \]  

where

\[ |x| = \sqrt{x^2 + y^2} \]

The \( L_{\text{mask}} \) is defined as the average binary cross-entropy loss using a per-pixel sigmoid function:

\[ L_{\text{mask}} = -\frac{1}{n} \sum_{i=1}^{n} y_i \log y'_i + (1 - y_i) \log (1 - y'_i) \]  

4. Experiments

In this section, experiments with the proposed ISASDNet are conducted on two datasets. Our network is compared with the other state of the art deep learning algorithms. The Common Objects in Context (COCO) metrics are used to evaluate performance. Two traditional algorithms are also compared with our algorithm. Finally, the proposed modules are analyzed, and several groups of performance comparison experiments are carried out.

4.1. Datasets and Evaluation Metrics

In order to promote the development of object detection in SAR images, Wang et al. constructed a ship detection dataset called SAR-Ship-Dataset [59]. In this dataset, 102 Gaofen-
3 images and 108 Sentinel-1 images are divided into 43,819 image chips with a length and width of 256. All image chips are labeled by SAR experts according to the Pascal Visual Object Classes (PASCAL VOC) standard. The resolution of these images involves 3 m, 5 m, 8 m and 10 m with different imaging modes. Moreover, these images contain complex environments, such as ports, inshore waters, and islands, and the ships are distributed in many forms, including independent cruise and fleet navigation. The top row of Figure 7 shows some example images from this dataset.

Figure 7. Some samples from two datasets. (a–j) The top row shows some samples from SAR-Ship-Dataset. (k–r) The bottom row presents some samples from SSDD. Both datasets contain a variety of complex scenarios, such as ports, inshore waters, and islands.

The other dataset used in our experiments is the SAR ship detection dataset (SSDD) [60]. The SSDD dataset also follows the PASCAL VOC standard. SSDD contains all kinds of SAR images with different polarizations, and sea conditions. There are 1160 images and 2540 ships in SSDD. The resolutions of SAR images are 1 m, 3 m, 5 m, 7 m, and 10 m. The bottom row of Figure 7 shows some example images of SSDD.

COCO [61] metrics constitute a classic evaluation standard for object detection and image segmentation. In various object detection competitions, COCO metrics are often used to measure algorithm performance. The intersection over union (IOU) is the core of COCO metrics, and refers to the intersection ratio between the bounding box predicted by algorithms and the ground truth:

$$\text{IOU} = \frac{B_p \cap B_g}{B_p \cup B_g}$$  \hspace{1cm} (17)

where $B_p$ is the predicted result, and $B_g$ is the ground truth. According to the preset IOU threshold, the precision and recall rate can be calculated by

$$\text{precision} = \frac{TP}{TP + FP}, \quad \text{recall} = \frac{TP}{TP + FN}$$  \hspace{1cm} (18)

where $TP$ is the number of true positives, $FP$ is the number of false positives, and $FN$ is the number of false negatives. For single class target ship detection, the mean average precision (AP) is defined as follows [50]:

$$\text{AP} = \int_0^1 p(r) \, dr$$  \hspace{1cm} (19)

where $r$ represents recall and $p(r)$ denotes the precision value that $\text{recall} = r$ corresponds to. In COCO metrics, different IOU thresholds produce different AP values. Table 1 lists the COCO metrics.
### Table 1. COCO metrics.

| Metric   | Meaning                                                                                           |
|----------|---------------------------------------------------------------------------------------------------|
| AP       | AP at \(\text{IOU} = 0.50:0.05:0.95\)                                                            |
| \(\text{AP}_{50}\) | AP at \(\text{IOU} = 0.50\)                                                                        |
| \(\text{AP}_{75}\) | AP at \(\text{IOU} = 0.75\)                                                                        |
| \(\text{AP}_s\) | AP for small objects: area < 32\(^2\) (\(\text{IOU} = 0.50:0.05:0.95\))                           |
| \(\text{AP}_M\) | AP for medium objects: 32\(^2\) < area < 96\(^2\) (\(\text{IOU} = 0.50:0.05:0.95\))              |
| \(\text{AP}_L\) | AP for large objects: area > 96\(^2\) (\(\text{IOU} = 0.50:0.05:0.95\))                          |

The result of traditional algorithm is a segmentation image. The foreground is ship pixel, and the other is background pixel. We use a pixel level figure of merit (FoM) [26] to evaluate the traditional algorithms. FoM is defined as

\[
\text{FoM} = \frac{tp}{tp + fn + fp} \tag{20}
\]

where true positive (\(tp\)) denotes the number of pixels belonging to ships which is correctly detected, the false positive (\(fp\)) denotes the number of false alarm pixels, and the false negative (\(fn\)) denotes missing ship pixels. Ground truth and the output results of ISASDNet are boxes. We regard the pixels in the boxes as ship pixels and the pixels outside the boxes as background pixels.

### 4.2. Experiment Results

In order to verify the performance of ISASDNet, we conducted experiments on the two aforementioned datasets. We also compared ISASDNet with other algorithms: Faster R-CNN, Mask R-CNN, YOLOv3, YOLOv4 [33], SSD, M2Det [62], RefineDet [63], D2Det [64], cell averaging CFAR (CA-CFAR) [65], and visual attention model (VAM) [66]. CA-CFAR is an improved CFAR algorithm based on statistics. VAM uses saliency maps to cluster and extract ships. CA-CFAR and VAM are traditional algorithms, and the other algorithms are deep learning algorithms. When training the networks, 70% of the images were randomly selected to constitute the training set while the remaining 30% constituted the test set. All experiments were implemented in Python3.7 with a Quadro P5200 GPU.

Since the backbone networks have an important influence on the result of object detection, ISASDNet uses two different backbone networks to extract features, namely ResNet50 and ResNet101. Like ISASDNet, Faster R-CNN, and Mask R-CNN use ResNet50 and ResNet101 to act as backbone networks. Their backbone networks also include RPN and FPN. The mask that is extracted through the designed strategy is introduced during training of Mask R-CNN. VGG16 is the backbone network of M2Det and RefineDet.

#### 4.2.1. Results on SAR-Ship-Dataset

Figures 8 and 9 present the experimental results obtained by various deep learning algorithms on SAR-Ship-Dataset. The green boxes represent the ground truth, and the red boxes denote the predicted results.

Different sizes of ships, inshore scenes, and close arrangement of ships increase the difficulty of detection. Still, Faster R-CNN has good performance and can identify ships in different scenes. When the backbone network of Faster R-CNN is ResNet50, it has a lower false detection rate and missed detection rate, and higher ship detection confidence. When training Mask R-CNN, instance segmentation loss was introduced, which affects ship detection. For nearshore ships and dense ships, the detection results of Mask R-CNN are worse than those of Faster R-CNN. For example, Mask R-CNN identifies some nearshore buildings as ships. As the classical single-stage detection algorithm, YOLOv3 and SSD transform target detection into a regression problem. Their accuracies are lower than those of the two-stage algorithms. Both YOLOv3 and SSD have a lot of missed detection, and their recognition confidence is lower than that of Mask R-CNN. YOLOv4, which has better performance than YOLOv3, is a very advanced single-stage algorithm. The detection
results of YOLOv4 are very good, and there are few false checks. RefineDet, which is a variant of SSD, has a good recognition effect for ships located in open waters, but easily misses nearshore ships. Although M2Det also has good performance, it is easy to find that the recognition result of M2Det cannot completely include the ships. D2Det is a new two-stage detection method introducing a dense local regression to improve ship detection accuracy. However, there are still a few false checks in its results. The last two rows in Figure 9 present the results of our proposed ISASDNet. ISASDNet can detect ships well under various complicated conditions; moreover, ISASDNet has the fewest number of false detections and missed detections of all algorithms.

Figure 8. Experimental results in SAR-Ship-Dataset. (a1–a5) The first row contains ground truth images. (b1–b5) The second row contains the results of Faster R-CNN with ResNet50. (c1–c5) The third row contains the results of Faster R-CNN with ResNet101. (d1–d5) The fourth row contains the results of Mask R-CNN with ResNet50. (e1–e5) The fifth row contains the results of Mask R-CNN with ResNet101. (f1–f5) The sixth row contains the results of YOLOv3. (g1–g5) The seventh row contains the results of YOLOv4.
Figure 9. Experimental results on SAR-Ship-Dataset. (a1–a5) The first row contains ground truth images. (b1–b5) The second row contains the results of SSD. (c1–c5) The third row contains the results of RefineDet. (d1–d5) The fourth row contains the results of M2Det. (e1–e5) The fifth row contains the results of D2Det. (f1–f5, g1–g5) The last two rows contain the experimental results of our algorithm: the fifth row shows the results obtained by ISASDNet with ResNet50, and the sixth row shows the results obtained by ISASDNet with ResNet101.

The quantitative analysis results are shown in Table 2. Overall, ISASDNet with the ResNet50 backbone has the best performance, with an AP value of 0.601. The AP of ISASDNet with the ResNet101 backbone is 0.596, which is only 0.005 lower than the optimal highest value. YOLOv4 has the second-best detection performance. Its AP is 0.596. Among all the two-stage algorithms, D2Det has the highest AP value: 0.591. When the backbone network of Faster R-CNN is ResNet50 and ResNet101, the AP values are 0.586
and 0.569, respectively. The AP values of Mask R-CNN are lower than that of Faster R-CNN because of the influence of instance segmentation loss. The AP values of Mask R-CNN are 0.448 and 0.433 for, respectively, backbone networks ResNet50 and ResNet101. The results of the classical one-stage detection algorithms are worse. For instance, the AP values of YOLOv3 and SSD are 0.359 and 0.343, respectively. Although RefineDet and M2Det have better results than YOLOv3 and SSD, their AP values are only 0.450 and 0.372, respectively.

Table 2. AP of different methods on SAR-Ship-Dataset.

| Method           | AP  | AP50 | AP75 | AP5 | APM | AP1 |
|------------------|-----|------|------|-----|-----|-----|
| faster-rcnn + r50| 0.586| 0.944| 0.669| 0.557| 0.631| 0.646|
| faster-rcnn + r101| 0.569| 0.917| 0.627| 0.586| 0.543| 0.460|
| mask-rcnn + r50  | 0.448| 0.852| 0.540| 0.452| 0.439| 0.351|
| mask-rcnn + r101 | 0.433| 0.851| 0.543| 0.429| 0.441| 0.382|
| YOLOv3           | 0.359| 0.760| 0.356| 0.377| 0.339| 0.227|
| YOLOv4           | 0.596| 0.954| 0.669| 0.597| 0.589| 0.592|
| SSD              | 0.343| 0.755| 0.397| 0.327| 0.368| 0.325|
| RefineDet        | 0.450| 0.888| 0.572| 0.464| 0.427| 0.224|
| M2Det            | 0.372| 0.836| 0.413| 0.437| 0.312| 0.259|
| D2Det            | 0.591| 0.948| 0.671| 0.597| 0.587| 0.581|
| ISASDNet + r50   | 0.601| 0.953| 0.652| 0.615| 0.582| 0.544|
| ISASDNet + r101  | 0.596| 0.958| 0.694| 0.609| 0.587| 0.578|

AP50 and AP75 are also very important evaluation indices. Although the AP50 of all algorithms is greater than 0.750, that of ISASDNet is the best. Regardless of whether the backbone network is ResNet50 or ResNet101, the AP50 values of ISASDNet are higher than 0.95. Meanwhile, D2Det and ISASDNet with the ResNet101 backbone have the highest AP75 of all compared algorithms. Although their results are very close, the performance of ISASDNet is a little better than that of D2Det. Objects can be divided by their size into small object, medium object, and large object. In SAR-Ship-Dataset, small objects account for 60.0% of all objects, medium object accounts for 39.7%, and large objects only account for 0.3%. Therefore, the accurate detection of small and medium objects is more important. As can be seen from Table 2, ISASDNet has the largest AP5 value. The AP5 of ISASDNet with the ResNet50 backbone is 0.615, and the AP5 of ISASDNet with the ResNet101 backbone is 0.609. For medium objects, Faster R-CNN with the ResNet50 backbone has the best result, with an APM of 0.631. The proposed ISASDNet also has good results, the APM values of ISASDNet are higher than 0.58. Although ISASDNet is not as good as Faster R-CNN, it also has good results for large objects. In summary, Table 2 shows that ISASDNet has better performance than the other deep learning algorithms.

The proposed ISASDNet is also compared with the traditional algorithms CA-CFAR and VAM. When calculating FoM, the results of ISASDNet and ground truth are transformed into binary segmentation images. In other words, the pixels in the predicted boxes and the ground truth boxes are regarded as ship pixels, and other pixels outside boxes are background pixels. When ISASDNet produces prediction results, the confidence is set to 0.5. Figure 10 shows the results of CA-CFAR and VAM on the SAR-Ship-Dataset. Two traditional algorithms can extract the ships far away from the coast, but they are less effective in detecting ships near shore. CA-CFAR and VAM will mistakenly detect land as ships, and small ships cannot be extracted completely. Table 3 shows the FoM of each algorithm. The FoM of ISASDNet with ResNet101 backbone is highest. The FoM of CA-CFAR and VAM are 0.1103 and 0.1691. The performance of ISASDNet is much better than that of traditional algorithms.

Table 3. FoM on SAR-Ship-Dataset.

| Method         | CA-CFAR | VAM   | ISASDNet with ResNet50 | ISASDNet with ResNet101 |
|----------------|---------|-------|------------------------|-------------------------|
| FoM            | 0.1103  | 0.1691| 0.6287                 | 0.6515                  |
4.2.2. Results on SSDD

We also compared the performances of ISASDNet and other algorithms on the SSDD dataset. Figures 11 and 12 show the detection results of deep learning algorithms, and Table 4 presents the quantitative analysis. From the experimental results, it can be seen that YOLOv4 and D2Det detect ships very well. Faster R-CNN is slightly inferior to YOLOv4 and D2Det. Compared with Faster R-CNN, Mask R-CNN has a poor recognition result for closely arranged ships. Moreover, Mask R-CNN often has false detection on closely arranged ships and inshore ships. YOLOv3 and SSD have poor recognition results for small ships and nearshore ships, and often has missed detection. While the detection results of RefineDet and M2Det are slightly better than those of YOLOv3 and SSD, they also often make mistakes and omissions. ISASDNet achieves the best results compared with the other algorithms. Even in the complex offshore situation, ISASDNet has good detection results.

| Method                  | AP   | AP₅₀  | AP₇₅  | APₛ   | AP₇₅ | AP₁₅ |
|------------------------|------|-------|-------|-------|------|------|
| faster-rcnn + r50      | 0.587| 0.947 | 0.629 | 0.609 | 0.561| 0.477|
| faster-rcnn + r101     | 0.579| 0.956 | 0.618 | 0.593 | 0.552| 0.490|
| mask-rcnn + r50        | 0.557| 0.927 | 0.585 | 0.571 | 0.522| 0.489|
| mask-rcnn + r101       | 0.563| 0.921 | 0.589 | 0.588 | 0.525| 0.463|
| YOLOv3                 | 0.508| 0.909 | 0.546 | 0.534 | 0.458| 0.474|
| YOLOv4                 | 0.601| 0.960 | 0.674 | 0.616 | 0.594| 0.579|
| SSD                    | 0.481| 0.865 | 0.520 | 0.511 | 0.437| 0.407|
| RefineDet              | 0.588| 0.949 | 0.598 | 0.610 | 0.563| 0.530|
| M2Det                  | 0.498| 0.903 | 0.531 | 0.531 | 0.462| 0.410|
| D2Det                  | 0.594| 0.955 | 0.643 | 0.599 | 0.582| 0.586|
| ISASDNet + r50         | 0.610| 0.954 | 0.677 | 0.624 | 0.605| 0.552|
| ISASDNet + r101        | 0.627| 0.968 | 0.685 | 0.636 | 0.603| 0.525|

As can be seen from Table 4, all algorithms produce good results on the SSDD dataset. All AP values are higher than 0.480, all AP₅₀ are higher than 0.850, and all AP₇₅ are higher than 0.50. However, ISASDNet with the ResNet101 backbone has the best performance (with an AP value of 0.627); meanwhile, the AP value of ISASDNet with the ResNet50 backbone is 0.610. The AP values of Faster R-CNN are 0.587 and 0.579 with backbones ResNet50 and ResNet101, respectively. The APₛ of Mask R-CNN with ResNet50 and Mask R-CNN with ResNet101 are 0.557 and 0.563, respectively. The APₛ of YOLOv3, SSD,
RefineDet and M2Det are lower, hitting 0.508, 0.481, 0.588, and 0.498, respectively. The performance of YOLOv4 and D2Det is only a little worse than that of our algorithm, with AP values being 0.601 and 0.594. Through these results, it can be concluded that ISASDNet has better and more robust detection performance.

Figure 11. Experimental results on SSDD. (a1–a4) The first row contains ground truth images. (b1–b4) The second row contains the results of Faster R-CNN with ResNet50. (c1–c4) The third row contains the results of Faster R-CNN with ResNet101. (d1–d4) The fourth row contains the results of Mask R-CNN with ResNet50. (e1–e4) The fifth row contains the results of Mask R-CNN with ResNet101. (f1–f4) The sixth row contains the results of YOLOv3. (g1–g4) The seventh row contains the results of YOLOv4.
Figure 12. Experimental results on SSDD. (a1–a4) The first row contains ground truth images. (b1–b4) The second row contains the results of SSD. (c1–c4) The third row contains the results of RefineDet. (d1–d4) The fourth row contains the results of M2Det. (e1–e4) The fifth row contains the results of D2Det. The last two rows contain the experimental results of our algorithm: (f1–f4) the fifth row show the results obtained by ISASDNet with ResNet50, (g1–g4) and the sixth row shows the results obtained by ISASDNet with ResNet101.

Figure 13 shows the results of CA-CFAR and VAM on the SSDD. VAM can extract more complete ship contour than CA-CFAR, and reduce the false alarm rate. Both CA-CFAR and VAM are easy to mix coastal ships with land. In complex scenes, the detection rate of traditional algorithm is still relatively low. Table 5 shows the results of quantitative analysis on the SSD. ISASDNet is much better than CA-CFAR and VAM. The FoM of ISASDNet with ResNet101 backbone is 0.6632, which is the highest.
4.3. Discussion

4.3.1. Ablation Experiment and Parameter Analysis

The proposed GRM and MASDM have a great impact on ISASDNet. We conducted an ablation experiment with the GRM and MASDM, and analyzed the experimental results obtained in various situations. The proposed ISASDNet was based on Mask R-CNN. Thus, Mask R-CNN whose backbone was ResNet50 was regarded as the baseline (Case 1). For the GRM analysis, the mask branch of Mask R-CNN was replaced by the GRM module (Case 2). In order to evaluate the contribution of the MASDM to ship detection, the pixel branch of ISASDNet was replaced by the mask branch of Mask R-CNN (Case 3). In other words, the network in Case 3 had an extra MASDM module than the network in Case 1. Case 4 involves the complete ISASDNet. We carried out ship detection experiments on the SAR-Ship-Dataset. Table 6 shows the results obtained in the four cases.

Table 6. The results of four cases.

| Case   | AP  | AP$_{50}$ | AP$_{75}$ | AP$_{S}$ | AP$_{M}$ | AP$_{L}$ |
|--------|-----|-----------|-----------|----------|----------|----------|
| Case 1 | 0.448 | 0.852     | 0.540     | 0.452    | 0.439    | 0.351    |
| Case 2 | 0.461 | 0.845     | 0.525     | 0.476    | 0.450    | 0.448    |
| Case 3 | 0.567 | 0.924     | 0.611     | 0.538    | 0.582    | 0.471    |
| Case 4 | 0.601 | 0.953     | 0.652     | 0.615    | 0.593    | 0.544    |

As can be seen from Table 6, the results from Case 1 and Case 2 are similar. The AP value in Case 2 is only 0.013 higher than that in Case 1. In Case 1 and Case 2, ISASDNet has no MASDM. ISASDNet cannot fuse the features extracted from segmentation task and those extracted from the detection task. Thus, there is a big gap between the results obtained in these cases and those obtained in Case 3 and Case 4. The AP value in Case 3 is 0.106 higher than that in Case 2. The MASDM can interact with the information from the object branch and pixel branch to promote the final detection results. It is easy to conclude that the best result is obtained in Case 4. In Case 4, ISASDNet can extract the segmentation result better than the mask branch in Case 3. This makes the AP result in Case 4 higher than that in Case 3. Figure 14 shows the object detection and instance segmentation results.
obtained in Case 3 and Case 4. Although the segmentation results in these two cases cannot extract the ship contour very well, the segmentation results obtained in Case 4 are better than those obtained in Case 3. Thus, the GRM promotes the results of ship detection to a certain extent.

**Figure 14.** Object detection results and instance segmentation results. (a–e) The top row contains ground truth images. (f–j) The middle row contains the results of Case 3. (k–o) The bottom row contains the results of Case 4.

In the MASDM, the length \( m \) of two 1D convolution kernels is very important since it affects the probability value of each point. We used ISASDNet with different \( m \) values to train on SAR-Ship-Dataset. Figure 15 shows the results. When the length \( m \) of the convolution kernel is 3, the MASDM can achieve the best ship detection.

**Figure 15.** Results of ISASDNet with different length \( m \) of two 1D convolution kernels. The recognition accuracy of ISASDNet decreases with the increase in \( m \).

### 4.3.2. Performance Comparison

In this subsection, we compared the performance of ISASDNet and other deep learning algorithms from various aspects. First, we counted the inference time of each algorithm. Then, we used different amount of data to train each algorithm and compared their performance. Finally, we added noise to the test image to evaluate the robustness of these algorithms.
Table 7 shows the inference time of each algorithm. The inference time is obtained in a Quadro P5200 GPU. Obviously, the one-stage object detection algorithms are faster than the two-stage algorithms. The inference time of these one-stage algorithms is less than 0.1 s. Faster R-CNN, Mask R-CNN, D2Det, and the proposed ISASDNet are two-stage algorithms. The inference time of these algorithms will be different when using different backbone networks. The inference speed of ISASDNet is slower than other algorithms, because the well-designed GRM module and MASDM module increase the inference time.

Table 7. Inference time (second).

| Method          | Faster-rcnn + r50 | Faster-rcnn + r101 | Mask-rcnn + r50 | Mask-rcnn + r101 |
|-----------------|------------------|-------------------|----------------|-----------------|
| Time            | 0.22546          | 0.28357           | 0.23263        | 0.29853         |

Experiments with different amounts of data were verified on the SAR-Ship-Dataset. We used 55%, 60%, and 65% data to train the deep learning algorithms, respectively. Table 8 shows the AP of each algorithm under different data volume. From the three groups of experiments, ISASDNet has the best performance. Whether the backbone network is ResNet101 or ResNet50, the AP value of ISASDNet is higher than other algorithms. When the training data set is 55% of the total, the AP value of ISASDNet is not less than 0.58. Faster R-CNN, YOLOv4 and D2Det are slightly worse than ISASDNet. Their AP values are higher than 0.50. The performance of Mask R-CNN, YOLOv3, SSD, RefineDet, and M2Det are not satisfactory. Their AP values are lower than 0.50. With the increase of the amount of data, the accuracy of each algorithm is increasing. When the training data set is 60% of the total, ISASDNet can produce the highest AP value of 0.59. The performance of D2Det is better than other algorithms, but worse than ISASDNet. YOLOv4 and Faster R-CNN also have good performance, and their AP values are higher than 0.55. While the AP values of Mask R-CNN, YOLOv3, SSD, RefineDet, and M2Det are lower than 0.50. When the training data set is 65% of the total, the performance of ISASDNet is still better than other algorithms.

In order to verify the robustness of the deep learning algorithms, Gaussian noise was added to the test images. The three groups of noise were Gaussian noise with mean value of 0 and variance of 0.1, 0.2 and 0.3, respectively. Noise experiments were carried out on SAR-Ship-Dataset. 65% of the images were selected to constitute the training set. Figure 16 shows the ISASDNet with ResNet101 backbone detection results under different noises. The noise mainly affects the detection of small targets by ISASDNet. Many small ships have been missed, and the prediction box may be offset. Table 9 shows the AP values of each algorithm for images with different noises. In general, the greater the variance of the added noise, the lower the detection accuracy of the algorithms. When the variance of noise is 0.1, the AP value of ISASDNet with ResNet101 backbone is 0.559. When the variance of noise is 0.3, the AP value of ISASDNet with ResNet101 backbone is reduced to 0.518. The AP value of D2Det decreases from 0.546 to 0.508 with the increase of noise. Additionally, the AP value of YOLOv4 decreases from 0.546 to 0.503. While the accuracy of other algorithms will decrease more with the increase of noise. Throughout Table 9, we can conclude that the AP value of ISASDNet is higher than that of other algorithms under different noises. This also proves that ISASDNet has better robustness.
Table 8. AP of methods under different amount of data.

| Data Volume | Methods              | AP  | AP_{50} | AP_{75} |
|-------------|----------------------|-----|---------|---------|
| 55%         | faster-rcnn + r50    | 0.551 | 0.929  | 0.620  |
|             | faster-rcnn + r101   | 0.529 | 0.889  | 0.594  |
|             | mask-rcnn + r50      | 0.428 | 0.836  | 0.520  |
|             | mask-rcnn + r101     | 0.399 | 0.794  | 0.513  |
|             | YOLOv3               | 0.268 | 0.604  | 0.184  |
|             | YOLOv4               | 0.549 | 0.929  | 0.612  |
|             | SSD                  | 0.328 | 0.739  | 0.361  |
|             | RefineDet            | 0.429 | 0.849  | 0.537  |
|             | M2Det                | 0.354 | 0.787  | 0.382  |
|             | D2Det                | 0.551 | 0.919  | 0.621  |
|             | ISASDNet + r50       | 0.585 | 0.931  | 0.621  |
|             | ISASDNet + r101      | 0.585 | 0.939  | 0.651  |
| 60%         | faster-rcnn + r50    | 0.558 | 0.925  | 0.624  |
|             | faster-rcnn + r101   | 0.545 | 0.905  | 0.603  |
|             | mask-rcnn + r50      | 0.437 | 0.845  | 0.527  |
|             | mask-rcnn + r101     | 0.405 | 0.813  | 0.521  |
|             | YOLOv3               | 0.270 | 0.612  | 0.179  |
|             | YOLOv4               | 0.561 | 0.940  | 0.528  |
|             | SSD                  | 0.335 | 0.746  | 0.389  |
|             | RefineDet            | 0.439 | 0.856  | 0.551  |
|             | M2Det                | 0.365 | 0.809  | 0.394  |
|             | D2Det                | 0.566 | 0.925  | 0.640  |
|             | ISASDNet + r50       | 0.588 | 0.945  | 0.635  |
|             | ISASDNet + r101      | 0.590 | 0.944  | 0.676  |
| 65%         | faster-rcnn + r50    | 0.567 | 0.932  | 0.643  |
|             | faster-rcnn + r101   | 0.558 | 0.910  | 0.618  |
|             | mask-rcnn + r50      | 0.449 | 0.853  | 0.542  |
|             | mask-rcnn + r101     | 0.429 | 0.836  | 0.535  |
|             | YOLOv3               | 0.291 | 0.625  | 0.217  |
|             | YOLOv4               | 0.588 | 0.947  | 0.639  |
|             | SSD                  | 0.339 | 0.754  | 0.397  |
|             | RefineDet            | 0.447 | 0.868  | 0.562  |
|             | M2Det                | 0.370 | 0.821  | 0.403  |
|             | D2Det                | 0.583 | 0.933  | 0.647  |
|             | ISASDNet + r50       | 0.592 | 0.949  | 0.643  |
|             | ISASDNet + r101      | 0.596 | 0.951  | 0.684  |

Table 9. AP under different noises.

| Noise Variance | Methods              | AP  | AP_{50} | AP_{75} |
|----------------|----------------------|-----|---------|---------|
| 0.1            | faster-rcnn + r50    | 0.523 | 0.888  | 0.566  |
|                | faster-rcnn + r101   | 0.523 | 0.895  | 0.561  |
|                | mask-rcnn + r50      | 0.377 | 0.805  | 0.469  |
|                | mask-rcnn + r101     | 0.369 | 0.806  | 0.450  |
|                | YOLOv3               | 0.235 | 0.584  | 0.137  |
|                | YOLOv4               | 0.546 | 0.904  | 0.592  |
|                | SSD                  | 0.209 | 0.507  | 0.102  |
|                | RefineDet            | 0.389 | 0.814  | 0.497  |
|                | M2Det                | 0.365 | 0.796  | 0.461  |
|                | D2Det                | 0.546 | 0.902  | 0.585  |
|                | ISASDNet + r50       | 0.357 | 0.921  | 0.633  |
|                | ISASDNet + r101      | 0.559 | 0.928  | 0.648  |
Table 9. Cont.

| Noise Variance | Methods          | AP   | AP50  | AP75  |
|----------------|------------------|------|-------|-------|
|                | faster-rcnn + r50| 0.511| 0.881 | 0.540 |
| 0.2            | faster-rcnn + r101| 0.508| 0.893 | 0.524 |
|                | mask-rcnn + r50  | 0.365| 0.799 | 0.452 |
|                | mask-rcnn + r101| 0.362| 0.791 | 0.448 |
|                | YOLOv3           | 0.153| 0.404 | 0.088 |
|                | YOLOv4           | 0.513| 0.893 | 0.559 |
|                | SSD              | 0.236| 0.602 | 0.158 |
|                | RefineDet        | 0.408| 0.841 | 0.492 |
|                | M2Det            | 0.362| 0.802 | 0.466 |
|                | D2Det            | 0.526| 0.902 | 0.581 |
|                | ISASDNet + r50   | 0.538| 0.907 | 0.605 |
|                | ISASDNet + r101  | 0.542| 0.911 | 0.629 |
|                | faster-rcnn + r50| 0.491| 0.870 | 0.503 |
| 0.3            | faster-rcnn + r101| 0.482| 0.873 | 0.486 |
|                | mask-rcnn + r50  | 0.348| 0.782 | 0.425 |
|                | mask-rcnn + r101| 0.351| 0.782 | 0.434 |
|                | YOLOv3           | 0.169| 0.435 | 0.093 |
|                | YOLOv4           | 0.503| 0.873 | 0.497 |
|                | SSD              | 0.201| 0.563 | 0.139 |
|                | RefineDet        | 0.398| 0.821 | 0.454 |
|                | M2Det            | 0.341| 0.773 | 0.417 |
|                | D2Det            | 0.508| 0.868 | 0.500 |
|                | ISASDNet + r50   | 0.519| 0.886 | 0.588 |
|                | ISASDNet + r101  | 0.518| 0.891 | 0.593 |

Figure 16. The results of ISASDNet under different noises. (a–e) The first row of the images is added with Gaussian noise with a mean value of 0 and a variance of 0.1. (f–j) The second row of the images is added with Gaussian noise with a mean value of 0 and a variance of 0.2. (k–o) The third row of the images is added with Gaussian noise with a mean value of 0 and a variance of 0.3.
5. Conclusions

In this study, ISASDNet is proposed for SAR ship detection. ISASDNet, which has a two-branch structure, can use instance segmentation to promote ship detection. In the SAR image, the brightness of the ship is obviously higher than that of the sea surface. Therefore, the designed global relational inference layer maps features interaction space to learn the interaction between ship and background. A GRM based on global relational inference layers can extract the instance segmentation results of ships; meanwhile, the designed MASDM integrates the information of the object branch and the pixel branch to improve the accuracy of ship detection. We also design a strategy to extract the mask of the SAR ship to train ISASDNet. Experimental results on SAR-Ship-Dataset and the SSDD dataset prove that ISASDNet is better than other algorithms. Ablation experiments show that the GRM and MASDM can effectively improve the detection rate. In addition, the performance of ISASDNet is better than that of other algorithms when using different amount of data to train various algorithms. The noise experiments also proved ISASDNet is more robust than other algorithms.

In future work, we will focus on ship instance segmentation and ship detection for more complex remote sensing. We expect to apply graph convolution neural network, transformer, and semi supervised learning to SAR image. According to the characteristics of SAR image, we hope to design better object detection algorithm and semantic segmentation algorithm.

Author Contributions: Z.W. designed the method; Z.W. and B.R. performed the experiments; Z.W. analyzed the results and wrote the article; B.H. and Z.R. revised the paper; L.J. and S.W. gave some suggestions. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the Key Scientific Technological Innovation Research Project by Ministry of Education; the National Natural Science Foundation of China under Grant 61671350, 61771379, 61836009; the Foundation for Innovative Research Groups of the National Natural Science Foundation of China under Grant 61621005; the Key Research and Development Program in Shaanxi Province of China under Grant 2019ZDLGY03-05; 111 Project; the Fundamental Research Funds for the Central Universities under Grant XJS211904.

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. Wu, Z.; Hou, B.; Jiao, L. Multiscale CNN With Autoencoder Regularization Joint Contextual Attention Network for SAR Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2021**, *59*, 1200–1213. [CrossRef]
2. Leng, X.; Ji, K.; Zhou, S.; Xing, X. Ship Detection Based on Complex Signal Kurtosis in Single-Channel SAR Imagery. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 6447–6461. [CrossRef]
3. Chen, X.; Xiang, S.; Liu, C.; Pan, C. Vehicle Detection in Satellite Images by Hybrid Deep Convolutional Neural Networks. *IEEE Geosci. Remote Sens. Lett.* **2014**, *11*, 1797–1801. [CrossRef]
4. Zhou, H.; Wei, L.; Lim, C.P.; Creighton, D.; Nahavandi, S. Robust Vehicle Detection in Aerial Images Using Bag-of-words and Orientation Aware Scanning. *IEEE Trans. Geosci. Remote Sens.* **2018**, *56*, 7074–7085. [CrossRef]
5. 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]
6. Liu, N.; Cao, Z.; Cui, Z.; Pi, Y.; Dang, S. Multi-Scale Proposal Generation for Ship Detection in SAR Images. *Remote Sens.* **2019**, *11*, 526. [CrossRef]
7. Marino, A. A Notch Filter for Ship Detection with Polarimetric SAR Data. *IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens.* **2013**, *6*, 1219–1232. [CrossRef]
8. Wang, Y.; Liu, H. PolSAR Ship Detection Based on Superpixel-Level Scattering Mechanism Distribution Features. *IEEE Geosci. Remote Sens. Lett.* **2015**, *12*, 1780–1784. [CrossRef]
9. Lin, H.; Chen, H.; Wang, H.; Yin, J.; Yang, J. Ship Detection for PolSAR Images via Task-Driven Discriminative Dictionary Learning. *Remote Sens.* **2019**, *11*, 769. [CrossRef]
10. Robey, F.C.; Fuhrmann, D.R.; Kelly, E.J.; Nitzberg, R. A CFAR adaptive matched filter detector. IEEE Trans. Aerosp. Electron. Syst. 1992, 28, 208–216. [CrossRef]
11. He, K.; Zhang, X.; Ren, S.; Sun, J. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. IEEE Trans. Pattern Anal. Mach. Intell. 2015, 37, 1904–1916. [CrossRef] [PubMed]
12. Shin, H.C.; Roth, H.R.; Gao, M.; Lu, L.; Xu, Z.; Nogues, I.; Yao, J.; Mollura, D.; Summers, R.M. Deep Convolutional Neural Networks for Computer-Aided Detection: CNN Architectures, Dataset Characteristics and Transfer Learning. IEEE Trans. Med. Imaging 2016, 35, 1285–1298. [CrossRef] [PubMed]
13. Ji, S.; Xu, W.; Yang, M.; Yu, K. 3D Convolutional Neural Networks for Human Action Recognition. IEEE Trans. Pattern Anal. Mach. Intell. 2013, 35, 221–231. [CrossRef] [PubMed]
14. 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 (CVPR), Las Vegas, NV, USA, 27–30 June 2016.
15. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.; Berg, A.C. SSD: Single shot multibox detector. In Proceedings of the European Conference on Computer Vision (ECCV), Amsterdam, The Netherlands, 8–16 October 2016; pp. 21–37.
16. 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]
17. Girshick, R.B. Fast R-CNN. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
18. Cui, Z.; Li, Q.; Cao, Z.; Liu, N. Dense Attention Pyramid Networks for Multi-Scale Ship Detection in SAR Images. IEEE Trans. Geosci. Remote Sens. 2019, 57, 8983–8997. [CrossRef]
19. Wang, S.; Gong, Y.; Xing, J.; Huang, L.; Huang, C.; Hu, W. RDSNet: A New Deep Architecture for Reciprocal Object Detection and Instance Segmentation. In Proceedings of the Association for the Advancement of Artificial Intelligence (AAAI), New York, NY, USA, 7–12 February 2020; pp. 12208–12215.
20. He, K.; Gkioxari, G.; Dollar, P.; Girshick, R. Mask R-CNN. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 2980–2988.
21. Wang, Y.; Liu, H. A Hierarchical Ship Detection Scheme for High-Resolution SAR Images. IEEE Trans. Geosci. Remote Sens. 2012, 50, 4173–4184. [CrossRef]
22. Gao, G. A Parzen-Window-Kernel-Based CFAR Algorithm for Ship Detection in SAR Images. IEEE Geosci. Remote Sens. Lett. 2011, 8, 557–561. [CrossRef]
23. An, W.; Xie, C.; Yuan, X. An Improved Iterative Censoring Scheme for CFAR Ship Detection with SAR Imagery. IEEE Trans. Geosci. Remote Sens. 2014, 52, 4585–4595.
24. Li, T.; Liu, Z.; Xie, R.; Ran, L. An Improved Superpixel-Level CFAR Detection Method for Ship Targets in High-Resolution SAR Images. IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens. 2018, 11, 184–194. [CrossRef]
25. Gao, G.; Gao, S.; He, J.; Li, G. Adaptive Ship Detection in Hybrid-Polarimetric SAR Images Based on the Power–Entropy Decomposition. IEEE Trans. Geosci. Remote Sens. 2018, 56, 5394–5407. [CrossRef]
26. Lang, H.; Xi, Y.; Zhang, X. Ship Detection in High-Resolution SAR Images by Clustering Spatially Enhanced Pixel Descriptor. IEEE Trans. Geosci. Remote Sens. 2019, 57, 5407–5423. [CrossRef]
27. Pastina, D.; Fico, F.; Lombardo, P. Detection of ship targets in COSMO-SkyMed SAR images. In Proceedings of the IEEE Radar Conference (RADAR), Kansas City, MO, USA, 23–27 May 2011; pp. 928–933.
28. Le, Q.V.; Ngiam, J.; Coates, A.; Lahiri, A.; Prochnow, B.; Ng, A.Y. On optimization methods for deep learning. In Proceedings of the International Conference on Machine Learning (ICML), Bellevue, WA, USA, 28 June–2 July 2011; pp. 265–272.
29. Le, Q.V.; Zou, W.Y.; Yeung, S.Y.; Ng, A.Y. Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Colorado Springs, CO, USA, 21–23 June 2011; pp. 3361–3368.
30. Messina, M.; Greco, M.; Fabbri, L.; Pinelli, G. Modified Otsu’s algorithm: A new computationally efficient ship detection algorithm for SAR images. In Proceedings of the 6th Conference on Advances in Radar and Remote Sensing (TyWRRS), Naples, Italy, 12–14 September 2012; pp. 262–266.
31. Ouchi, K.; Tamaki, S.; Yaguchi, H.; Iehara, M. Ship Detection Based on Coherence Images Derived From Cross Correlation of Multilook SAR Images. IEEE Geosci. Remote Sens. Lett. 2004, 1, 184–187. [CrossRef]
32. Tello, M.; Martinez, C.L.; Mallorqui, J.J. A Novel Algorithm for Ship Detection in SAR Imagery Based on the Wavelet Transform. IEEE Geosci. Remote Sens. Lett. 2005, 2, 201–205. [CrossRef]
33. Bochkovskiy, A.; Wang, C.Y.; Liao, H. YOLOv4: Optimal Speed and Accuracy of Object Detection. arXiv 2020, arXiv:2004.10934v1.
34. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. In Proceedings of the International Conference on Learning Representations (ICLR), San Diego, CA, USA, 7–9 May 2015.
35. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
36. Huang, G.; Liu, Z.; Maaten, L.V.D.; Weinberger, K.Q. densely Connected Convolutional Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 22–25 July 2017; pp. 2261–2269.
37. 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]
38. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollar, P. Focal Loss for Dense Object Detection. *IEEE Trans. Pattern Anal. Mach. Intell.* 2020, 42, 318–327. [CrossRef]

39. Lin, T.Y.; Dollar, P.; Girshick, R.; He, K.; Harlighan, B.; Belongie, S. Feature Pyramid Networks for Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 22–25 July 2017; pp. 936–944.

40. Liu, S.; Qi, L.; Qin, H.; Shi, J.; Jia, J. Path Aggregation Network for Instance Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–22 June 2018; pp. 8759–8768.

41. Wu, F.; Zhou, Z.; Wang, B.; Ma, J. Inshore Ship Detection Based on Convolutional Neural Network in Optical Satellite Images. *IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens.* 2018, 11, 4005–4015. [CrossRef]

42. Yan, Y.; Tan, Z.; Su, N. A Data Augmentation Strategy Based on Simulated Samples for Ship Detection in RGB Remote Sensing Images. *ISPRS Int. J. Geo-Inf.* 2019, 8, 276. [CrossRef]

43. Li, L.; Zhou, Z.; Wang, B.; Miao, L.; Zong, H. A Novel CNN-Based Method for Accurate Ship Detection in HR Optical Remote Sensing Images with Rotated Bounding Box. *IEEE Trans. Geosci. Remote Sens.* 2021, 59, 686–699. [CrossRef]

44. Liu, Q.; Xiang, X.; Yang, Z.; Hu, Y.; Hong, Y. Arbitrary Direction Ship Detection in Remote-Sensing Images Based on Multitask Learning and Multiregion Feature Fusion. *IEEE Trans. Geosci. Remote Sens.* 2021, 59, 1553–1564. [CrossRef]

45. Ma, J.; Zhou, Z.; Wang, B.; Zong, H.; Wu, F. Ship Detection in Optical Satellite Images via Directional Bounding Boxes Based on Ship Center and Orientation Prediction. *Remote Sens.* 2019, 11, 2173. [CrossRef]

46. Feng, Y.; Diao, W.; Sun, X.; Yan, M.; Gao, X. Towards Automated Ship Detection and Category Recognition from High-Resolution Aerial Images. *Remote Sens.* 2019, 11, 1901. [CrossRef]

47. Fan, W.; Zhou, F.; Bai, X.; Tao, M.; Tan, T. Ship Detection Using Deep Convolutional Neural Networks for PolSAR Images. *Remote Sens.* 2019, 11, 2862. [CrossRef]

48. Fan, Q.; Chen, F.; Cheng, M.; Lou, S.; Xiao, R.; Zhang, B.; Wang, C.; Li, J. Ship Detection Using a Fully Convolutional Network with Compact Polarimetric SAR Images. *Remote Sens.* 2019, 11, 2171. [CrossRef]

49. Gao, F.; Shi, W.; Wang, J.; Yang, E.; Zhou, H. Enhanced Feature Extraction for Ship Detection from Multi-Resolution and Multi-Scene Synthetic Aperture Radar (SAR) Images. *Remote Sens.* 2019, 11, 2694. [CrossRef]

50. Wei, S.; Su, H.; Ming, J.; Wang, C.; Yan, M.; Kumar, D.; Shi, J.; Zhang, X. Precise and Robust Ship Detection for High-Resolution SAR Imagery Based on HR-SDNet. *Remote Sens.* 2020, 12, 167. [CrossRef]

51. Chen, C.; He, C.; Hu, C.; Pei, H.; Jiao, L. MSARN: A Deep Neural Network Based on an Adaptive Recalibration Mechanism for Multiscale and Arbitrary-Oriented SAR Ship Detection. *IEEE Access* 2019, 7, 159262–159283. [CrossRef]

52. Hou, B.; Ren, Z.; Zhao, W.; Wu, Q.; Jiao, L. Object Detection in High-Resolution Panchromatic Images Using Deep Models and Spatial Template Matching. *IEEE Trans. Geosci. Remote Sens.* 2019, 58, 956–970. [CrossRef]

53. Kang, M.; Leng, X.; Lin, Z.; Ji, K. 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.

54. Zou, L.; Zhang, H.; Wang, C.; Wu, F.; Gu, F. MW-ACGAN: Generating Multiscale High-Resolution SAR Images for Ship Detection. *Sensors* 2020, 20, 6673. [CrossRef]

55. Ridler, T.W.; Calvard, S. Picture Thresholding Using an Iterative Selection Method. *IEEE Trans. Syst.* 1978, 8, 630–632.

56. Monsalve, A.F.T.; Medina, J.V. Hardware implementation of ISODATA and Otsu thresholding algorithms. In Proceedings of the 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, Lyon, France, 22–26 August 2007.

57. Hou, B.; Yang, W.; Wang, S.; Hou, X. SAR image ship detection based on visual attention model. In Proceedings of the 2013 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), Melbourne, Australia, 21–26 July 2013.