sensors

MDPI

Article

MW-ACGAN: Generating Multiscale High-Resolution SAR Images for Ship Detection

Lichuan Zou [1,2], Hong Zhang [1,*], Chao Wang [1,2], Fan Wu [1] and Feng Gu [1,2]

[1] Key Laboratory of Digital Earth Science, Aerospace information Research Institute,
Chinese Academy of Sciences, Beijing 100094, China; zoulichuan20@mails.ucas.ac.cn (L.Z.);
wangchao@radi.ac.cn (C.W.); wufan@radi.ac.cn (F.W.); gufeng@radi.ac.cn (F.G.)

[2] College of Resources and Environment, University of Chinese Academy of Sciences, Beijing 100049, China

[*] Correspondence: zhanghong@radi.ac.cn; Tel.: +86-10-8217-8186

Received: 6 October 2020; Accepted: 19 November 2020; Published: 21 November 2020

check for updates

Abstract: In high-resolution Synthetic Aperture Radar (SAR) ship detection, the number of SAR samples seriously affects the performance of the algorithms based on deep learning. In this paper, aiming at the application requirements of high-resolution ship detection in small samples, a high-resolution SAR ship detection method combining an improved sample generation network, Multiscale Wasserstein Auxiliary Classifier Generative Adversarial Networks (MW-ACGAN) and the Yolo v3 network is proposed. Firstly, the multi-scale Wasserstein distance and gradient penalty loss are used to improve the original Auxiliary Classifier Generative Adversarial Networks (ACGAN), so that the improved network can stably generate high-resolution SAR ship images. Secondly, the multi-scale loss term is added to the network, so the multi-scale image output layers are added, and multi-scale SAR ship images can be generated. Then, the original ship data set and the generated data are combined into a composite data set to train the Yolo v3 target detection network, so as to solve the problem of low detection accuracy under small sample data set. The experimental results of Gaofen-3 (GF-3) 3 m SAR data show that the MW-ACGAN network can generate multi-scale and multi-class ship slices, and the confidence level of ResNet18 is higher than that of ACGAN network, with an average score of 0.91. The detection results of Yolo v3 network model show that the detection accuracy trained by the composite data set is as high as 94%, which is far better than that trained only by the original SAR data set. These results show that our method can make the best use of the original data set, improve the accuracy of ship detection.

Keywords: high-resolution SAR; ship detection; MW-ACGAN; Yolo v3

1. Introduction

Ship detection is an important maritime management technology, including the investigation of illegal fishing areas, oil spill detection, maritime traffic management, and national defense [1–9]. Synthetic aperture radar (SAR) is an active microwave imaging sensor whose all-day and all-weather working capacity give it an important place in marine exploration [4–9].

With the rise of artificial intelligence, deep learning provides powerful power for SAR ship detection and classification [10–24]. Deep learning, with its advantages of high accuracy, fast speed and less human intervention, has gradually become the dominant position in the field of SAR ship detection and classification [10–24]. Research has shown that deep learning detection algorithms with stronger generalization capabilities require more patterns of labeled data as support. However, because SAR images are expensive to obtain and difficult to interpret, it is an expensive and time-consuming task to create big data sets of SAR images with different scales. The biggest weakness of lacking SAR ship learning data is bound to cause the insufficiency of the above advantages, because deep learning always needs much labeled data to enrich the learning experience [11]. Some scholars, e.g., Wang et al. [11], Li et al. [24], Sun et al. [25] and Wei et al. [26], released their SAR ship detection datasets, but high resolution SAR ship labeled data sets are still lacking. With the development of SAR imaging technology to high resolution, there is an urgent need for a large number of free and public SAR data sets to adapt to high-resolution SAR ship detection.

SAR image simulation is the main method for overcoming the lack of high-resolution SAR data [27]. Traditional SAR image simulation tools are mainly divided into two categories: signal simulation and image simulation. Signal-based simulation is mostly based on electromagnetic theory to calculate the original radar echo signal and simulate microwave scattering processes, such as the Kirchhoff physical optics method [28], geometric optics approximation [29], integral equation method [30] and Phong model [31]. Image simulation algorithms are mostly based on geometric calculation, using gridding algorithms and ray tracing to approximate the electromagnetic propagation process. Such methods include the grid simulation software SARViz [32], and the ray tracing simulation software RaySAR [33] and CohRaS [34]. However, the calculation processes for these methods are complex, consume much computational memory, and cost a large amount of resources [35]. Moreover, these methods lack real data information in the simulation process, resulting in generated images with insufficient realism.

In recent years, the generative adversarial network (GAN) has achieved outstanding performance in computer vision [36–41]. The image generation algorithm based on GAN can generate realistic images from real image data, with low loss and end-to-end advantages. It is widely used in image generation [36–38], style transfer [39], super-resolution reconstruction [40,41] and other fields. Many researchers have noticed that the GAN can solve the problem of insufficient SAR data. Ren [27] et al. added the multi-scale structural similarity(MS-SSIM) loss item to the original Auxiliary Classifier Generative Adversarial Networks (ACGAN) loss function proposed by Augustus Odena [42] to generate 28 × 28 SAR land-target objects. Jiayi Guo added an angular loss item to the original GAN network loss function to generate 64 × 64 SAR tank targets with different angles [43]. Schwegmann used the Information Maximizing Generative Adversarial Network (InfoGAN) to generate low-resolution SAR ship targets [44]. Marmanis et al. used the Boundary Equilibrium Generative Adversarial Network (BEGAN) to generate SAR building targets of size 160 × 160 [45]. However, although these works propose the use of the GAN to generate SAR target objects, these methods have some shortcomings. When the network generates high-resolution images, training collapse often occurs due to unstable network gradient updates. It is necessary to constantly adjust parameters to find the optimal solution for generating high-resolution images. At the same time, the ACGAN does not impose constraints on the generated images of different scales. To generate images of different scales, the network structure needs to be changed constantly, which greatly increases the consumption of computing resources. In addition, the generated SAR images are only qualitatively evaluated visually, its practicability is not evaluated in scenarios.

Therefore, we need to conduct in-depth research on two issues. The first issue is how to use GANs to generate high-resolution SAR images of multiple scales stably to solve the bottleneck problem of insufficient SAR data. The second is whether the SAR images generated by GANs can be used in target detection frameworks to improve the accuracy of target detection.

Based on the above, this paper proposes an integrated framework of sample generation and detection in the case of small samples of high-resolution SAR ships. Firstly, an improved ACGAN method called Multiscale Wasserstein Auxiliary Classifier Generative Adversarial Networks (MW-ACGAN) is proposed, which uses the multi-scale Wasserstein distance and gradient penalty to make the network more stable, thus generating multiscale high-resolution SAR ship images. Then, the generated images are combined with the original small samples to train the Yolo v3 model to achieve high-precision ship detection under small samples. The remaining chapters of this paper are arranged as follows: the second section mainly introduces the proposed method, and the third section includes the experiments and results. The fourth part is a summary.

2. Method

Figure 1 shows the flow chart of this study, mainly including two processes. The first part is to generate high-resolution SAR ship images using the MW-ACGAN network. The MW-ACGAN backbone network consists of two parts, one is a generator and the other is a discriminator. The generator obtains images by random vector sampling and bilinear interpolation. The main purpose of the discriminator is to distinguish whether the image comes from the generated image or the real image. The multi-scale loss term is used to transfer the ship image generated by each scale to the discriminator of the corresponding scale, so as to output high-resolution ship images of different scales in real time. The second part is to train the Yolo v3 model by combining the generated high-resolution SAR ship data with the real SAR ship data. Then, some SAR images are used to evaluate the detection performance of the trained Yolo v3 model.

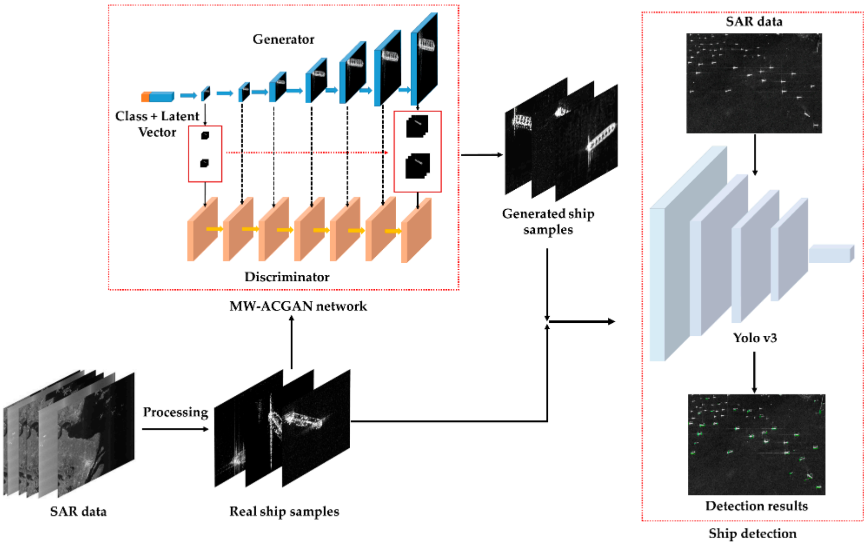


Figure 1. Research flow chart.

2.1. MW-ACGAN Network

2.1.1. GAN and ACGAN

Goodfellow [46] et al. proposed the GAN network, using two kinds of network structure model, generator and discriminator. The generator is used to simulate the data distribution of the real image, and the discriminator is used to judge whether the image is a generated image or a real image. Compared with the image generation model based on maximum likelihood function theory, GAN uses a neural network to simulate more complex functions, and generates higher dimensional images.

The purpose of the generator is to fit the distribution of the real sample data, while the purpose of the discriminator is to distinguish the generated sample from the real sample. The generator G takes the random noise vector z as input and finally outputs the image $X_{fake} = G(z)$. The purpose of the generator is to make the generated distribution X_{fake} fit the real data distribution X_{real}. The discriminator D inputs the generated image and the real image, and the output probability distribution of the image source $P(S|X) = D(X)$. The purpose of the discriminator is to distinguish between the generated image and the real image. The final generator and discriminator are opponents of the "game," which includes the minmax adversarial training method and alternate training at the same time. The optimization function is as follows,

$$\min_{G}\max_{D} V(D,G) = E_{x \sim X_{real}(x)}[\log D(x)] + E_{z \sim X_{fake}(z)}[\log(1 - D(G(z)))] \tag{1}$$

The optimization objective function is mainly divided into two parts: the optimization for generator G and the optimization for discriminator D. The optimization of discriminator D is shown as,

$$\begin{aligned} \mathrm{D}^{*} &= \underset{\mathrm{D}}{\mathrm{argmax}}\mathrm{V}(\mathrm{D},\mathrm{G}) \\ &= \underset{\mathrm{D}}{\mathrm{argmax}}\Big(\mathrm{E}_{\mathrm{x}\sim\mathrm{X}_{\mathrm{real}}}[\log\mathrm{D}(\mathrm{x})] + \mathrm{E}_{\mathrm{x}\sim\mathrm{X}_{\mathrm{fake}}}[\log(1-\mathrm{D}(\mathrm{x}))]\Big) \\ &= \underset{\mathrm{D}}{\mathrm{argmax}}\Big(\int \mathrm{X}_{\mathrm{real}}(\mathrm{x})\log\mathrm{D}(\mathrm{x})\mathrm{dx} + \int \mathrm{X}_{\mathrm{fake}}(\mathrm{x})\log(1-\mathrm{D}(\mathrm{x}))\mathrm{dx}\Big) \\ &= \underset{\mathrm{D}}{\mathrm{argmax}}\Big(\int [\mathrm{X}_{\mathrm{real}}(\mathrm{x})\log\mathrm{D}(\mathrm{x}) + \mathrm{X}_{\mathrm{fake}}(\mathrm{x})\log(1-\mathrm{D}(\mathrm{x}))]\mathrm{dx}\Big) \end{aligned} \tag{2}$$

The differential equation is obtained,

$$\mathrm{D}^{*}(\mathrm{x}) = \frac{\mathrm{X}_{\mathrm{real}}(\mathrm{x})}{\mathrm{X}_{\mathrm{real}}(\mathrm{x}) + \mathrm{X}_{\mathrm{fake}}(\mathrm{x})} \tag{3}$$

The optimal formula of discriminator D is obtained,

$$\begin{aligned} &\underset{\mathrm{D}}{\max}\mathrm{V}(\mathrm{G},\mathrm{D}) = \mathrm{V}(\mathrm{G},\mathrm{D}^{*}) \\ &= \mathrm{E}_{\mathrm{x}\sim\mathrm{P}_{\mathrm{data}}}\left[\log\frac{\mathrm{X}_{\mathrm{real}}(\mathrm{x})}{\mathrm{X}_{\mathrm{real}}(\mathrm{x}) + \mathrm{X}_{\mathrm{fake}}(\mathrm{x})}\right] + \mathrm{E}_{\mathrm{x}\sim\mathrm{P}_{\mathrm{G}}}\left[\frac{\mathrm{X}_{\mathrm{real}}(\mathrm{x})}{\mathrm{X}_{\mathrm{real}}(\mathrm{x}) + \mathrm{X}_{\mathrm{fake}}(\mathrm{x})}\right] \\ &= \int \mathrm{X}_{\mathrm{real}}(\mathrm{x})\log\left(\frac{\mathrm{X}_{\mathrm{real}}(\mathrm{x})}{\mathrm{X}_{\mathrm{real}}(\mathrm{x}) + \mathrm{X}_{\mathrm{real}}(\mathrm{x})}\right)\mathrm{dx} + \int \mathrm{X}_{\mathrm{fake}}(\mathrm{x})\log\left(\frac{\mathrm{X}_{real}(\mathrm{x})}{\mathrm{X}_{\mathrm{real}}(\mathrm{x}) + \mathrm{X}_{\mathrm{fake}}(\mathrm{x})}\right)\mathrm{dx} \\ &= -2\log 2 + \int \mathrm{X}_{\mathrm{real}}(\mathrm{x})\log\frac{\mathrm{X}_{\mathrm{real}}(\mathrm{x})}{(\mathrm{X}_{\mathrm{real}}(\mathrm{x}) + \mathrm{X}_{\mathrm{fake}}(\mathrm{x}))/2}\mathrm{dx} \\ &= -2\log 2 + \mathrm{KL}\Big(\mathrm{X}_{real}\|\frac{\mathrm{X}_{\mathrm{real}} + \mathrm{X}_{\mathrm{fake}}}{2}) + \mathrm{KL}(\mathrm{X}_{\mathrm{real}}\|\frac{\mathrm{X}_{\mathrm{real}} + \mathrm{X}_{\mathrm{fake}}}{2}\Big) \\ &= -2\log 2 + 2\mathrm{JSD}(\mathrm{X}_{\mathrm{real}}\|\mathrm{X}_{\mathrm{fake}}) \end{aligned} \tag{4}$$

From (4), the optimization goal of GAN discriminator D is to measure the JS divergence between the distribution $\mathrm{X}_{\mathrm{real}}$ of real sample data and the distribution $\mathrm{X}_{\mathrm{fake}}$ of generated sample data. An optimization for generator G is shown as,

$$\mathrm{G}^{*} = \underset{\mathrm{G}}{\mathrm{argmin}}\underset{\mathrm{D}}{\max}\mathrm{V}(\mathrm{G},\mathrm{D}) \tag{5}$$

From (5), the optimization goal of GAN generator G is to reduce the JS divergence between the distribution $\mathrm{X}_{\mathrm{fake}}$ of generated sample data and the distribution $\mathrm{X}_{\mathrm{real}}$ of real sample data.

However, the original GAN has difficulty generating a multicategory and high-quality data set [47]. The ACGAN proposed by Odena [42] realized the generation of 128 × 128 high-resolution images by adding the discrimination structure of image categories. The ACGAN adds a conditional category vector c based on the input noise z to constrain the generator and then uses an auxiliary classifier to provide the GAN with the ability to generate high-quality images of multiple categories. Due to the addition of the category discrimination layer, the final objective function contains two parts: the true and false discrimination probability L_{s} and the category discrimination probability L_{c}.

$$\mathrm{L}_{\mathrm{S}} = \mathrm{E}[\log\mathrm{P}(\mathrm{S} = \mathrm{real}|\mathrm{X}_{\mathrm{real}})] + \mathrm{E}[\log\mathrm{P}(\mathrm{S} = \mathrm{fake}|\mathrm{X}_{\mathrm{fake}})] \tag{6}$$

$$\mathrm{L}_{\mathrm{C}} = \mathrm{E}[\log\mathrm{P}(\mathrm{C} = \mathrm{c}|\mathrm{X}_{\mathrm{real}})] + \mathrm{E}[\log\mathrm{P}(\mathrm{C} = \mathrm{c}|\mathrm{X}_{\mathrm{fake}})] \tag{7}$$

The optimization goal of the discriminator D is to maximize $\mathrm{L}_{\mathrm{S}} + \mathrm{L}_{\mathrm{C}}$, and the optimization goal of the generator G is to maximize $\mathrm{L}_{\mathrm{S}}-\mathrm{L}_{\mathrm{C}}$. After continuous iteration, the ACGAN can generate high-quality images in multiple categories. However, when the ACGAN generates images with different resolutions, it is often necessary to change the network structure by terminating training, which greatly increases the consumption of computing resources. At the same time, when generating high-resolution images (such as 256 × 256 images), the problems of unstable gradient updates and mode collapses often occur due to the use of the JS divergence to measure data distribution differences. To solve the above problems, we design the following network structure.

2.1.2. MW-ACGAN

The proposed MW-ACGAN network structure is shown in Figure 2.

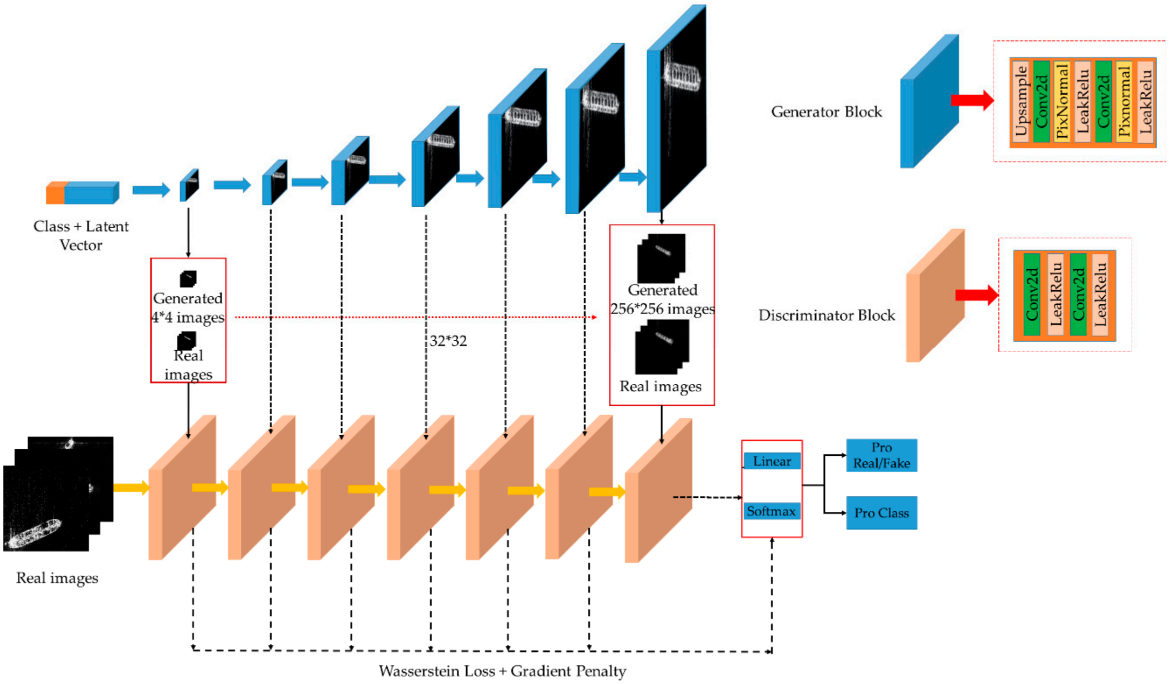


Figure 2. Our proposed network structure.

Figure 2 shows the detailed architecture of the proposed MW-ACGAN network. For the generator, as the blue block unit in the upper right of Figure 2, we use an upsampling layer, two convolutional layers, two local response normalization layers and two LeakyReLU activation functions. For the discriminator, we use two convolutional layers and two LeakyReLU activation functions to form a unit module.

We have made two improvements to the original ACGAN network, one is to use Wasserstein distance and gradient penalty terms to make training more stable. The other is to add multi-scale loss term to generate high-quality images of different scales. These two improvements are detailed below.

Wasserstein Distance and Gradient Penalty Terms

The ACGAN uses JS divergence to measure the differences between data distributions,

$$\mathrm{JS}(\mathrm{X}_{\mathrm{real}}\|\mathrm{X}_{\mathrm{fake}})=\frac{1}{2}\mathrm{KL}(\mathrm{X}_{\mathrm{real}}\|\frac{\mathrm{X}_{\mathrm{real}}+\mathrm{X}_{\mathrm{fake}}}{2})+\frac{1}{2}\mathrm{KL}(\mathrm{X}_{\mathrm{real}}\|\frac{\mathrm{X}_{\mathrm{real}}+\mathrm{X}_{\mathrm{fake}}}{2}) \tag{8}$$

where, $\mathrm{X}_{\mathrm{real}}$ is the probability distribution of real data, and $\mathrm{X}_{\mathrm{fake}}$ is the probability distribution of generated data. When JS divergence is used to measure the difference of data distribution, the following problems occur: minimizing the generator loss is equivalent to reducing the JS divergence of P_{r} and P_{g}, and in a high-dimensional space, the real data distribution and the generated data distribution usually do not coincide. Therefore, the JS divergence metrics are log2, which causes the gradient of the generator to be 0. Therefore, when the discriminator is overtrained, the generator tends to make the gradient disappear, which makes it difficult to decrease the loss of the generator. When the discriminator is undertrained, the generator will experience unstable gradient updates.

When the Wasserstein distance is used to measure the difference between data distributions, it is defined as follows,

$$W(X_{real}, X_{fake}) = \inf_{\gamma\sim\prod(X_{real},X_{fake})} E_{(x,y)}[\|x - y\|] \tag{9}$$

Even if the two distributions do not overlap in high-dimensional space, the Wasserstein distance can still measure the difference between them so that the problem of the disappearance of the gradient can be solved. However, because the $\inf_{\gamma\sim\prod(P_r,P_g)}$ method in (9) can not be solved directly, the Lipschitz transformation is used to transform the formula:

$$W(X_{real}, X_{fake}) = \frac{1}{K}\sup_{\|f\|_L\le K} E_{x\sim X_{real}}[f(x)] - E_{x\sim X_{fake}}[f(x)] \tag{10}$$

In this case, ω is used to define a series of possible functions $f_w(x)$. Then, (10) can be solved by

$$K\,W(X_{real}, X_{fake}) = \max_{w::|f_w|_L\le K} E_{x\sim X_{real}}[f_w(x)] - E_{x\sim X_{fake}}[f_w(x)] \tag{11}$$

A parameter W neural network f_w is constructed to maximize (12). At this time, L is approximately the Wasserstein distance between the real sample data and the generated sample data.

$$L = E_{x\sim X_{real}}[f_w(x)] - E_{x\sim X_{fake}}[f_w(x)] \tag{12}$$

Finally, the gradient penalty term is added, because the gradient loss term can effectively limit the gradient update range and prevent gradient explosion. The final loss function is as follows:

$$L'_S = E(\|D(X_S|S = real) - D(X_S|S = fake)\|) + \lambda\cdot[(\|\nabla_{\hat{z}}D(\hat{z})\|_2 - 1)^2] \tag{13}$$

In (13), the first term is the Wasserstein loss, and the second term is the gradient penalty term. The penalty coefficient λ is set to 10. The optimization goal of the final discriminator is to maximize $L'_S + L_C$, and the optimization goal of the generator is to maximize $L'_S - L_C$.

Multiscale Loss

Different scenarios have different resolution requirements, so it is an urgent need for a generation network that can generate multiple scales of SAR images at the same time to solve the problem of insufficient data. However, the original ACGAN did not impose constraints on the generation of images at different scales in the generation process, which made it difficult to adapt to the simultaneous generation of multiscale images. To generate multiple-scale images, the network structure needs to be changed and retrained, which greatly increases the amount computing resources consumed.

To solve this problem, we first establish a multiscale data set and downsample the real image size to 256 × 256 at different resolutions. Second, for the generator, we add a multiscale image output layer, that is, use 1 × 1 size convolution to perform channel compression on feature maps of different sizes (such as 4 × 4, 8 × 8, etc.), and then compress them to 3-channel feature maps at the corresponding scale. Next, we generate a 3-channel image and output it to the corresponding structural level discriminator, as shown by the black dotted line in the middle of Figure 2. Finally, we add a multiscale loss term to constrain the network to generate high-quality images with different scales. That is, we add Wasserstein loss and gradient penalty terms of multiple scales, update θ_G and θ_D through back propagation and add the losses of different scales.

$$L'_{S\_totall} = L'_{S\_4\times4} + L'_{S\_8\times8} + L'_{S\_16\times16} + L'_{S\_32\times32} + L'_{S\_64\times64} + L'_{S\_128\times128} + L'_{S\_256\times256} \tag{14}$$

$$L_{C\_totall} = L_{C\_4\times4} + L_{C\_8\times8} + L_{C\_16\times16} + L_{C\_32\times32} + L_{C\_64\times64} + L_{C\_128\times128} + L_{C\_256\times256} \tag{15}$$

The optimization goal of the discriminator D is to maximize $L'_{S\_totall} + L_{C\_totall}$, and the optimization goal of the generator is to maximize $L'_{S\_totall} - L_{C\_totall}$. During the training process, train the discriminator five times and then train the generator once. Finally, through continuous adversarial training, the network can adapt to the task of generating multiscale high-quality images.

2.2. Ship Detection Using Yolo v3 Model and Composite Dataset

The Yolo v3 [48–53] network was selected to test whether the SAR samples generated by our method can enhance the detection sample set.

Yolo v3 is an improvement based on Yolo v2, which is faster and more accurate. Its backbone network is darknet53. Yolo v3 will generate three different scale feature maps, which correspond to 1/32, 1/16 and 1/8 of the original size. Each scale corresponds to N channels, which contain the prediction information, the prediction results of each grid and the prior frames of each size. Yolo v3 uses the K-means clustering method to set three kinds of anchors for each subsampling scale, and nine kinds of anchors are clustered. Finally, the prediction frame is drawn on the predicted ship.

The flowchart of ship detection using the Yolo v3 model and the composite dataset is shown in Figure 3. The SAR image to be detected is segmented by a chessboard with the size of S × S, and then predict each sub image in SAR image by sliding window. Then, the sub images are resampled as the input size of the network, and input into the trained Yolo v3 network. Basic features are extracted from the backbone network, and multi-scale features can be extracted by fusing extra convolution layer features. Using the network calculation, the position parameters and labels of the prediction box on sub images are output. Finally, through coordinate conversion, the coordinates of the prediction frame are mapped to the original image, and the prediction results of all sub images are integrated and output.

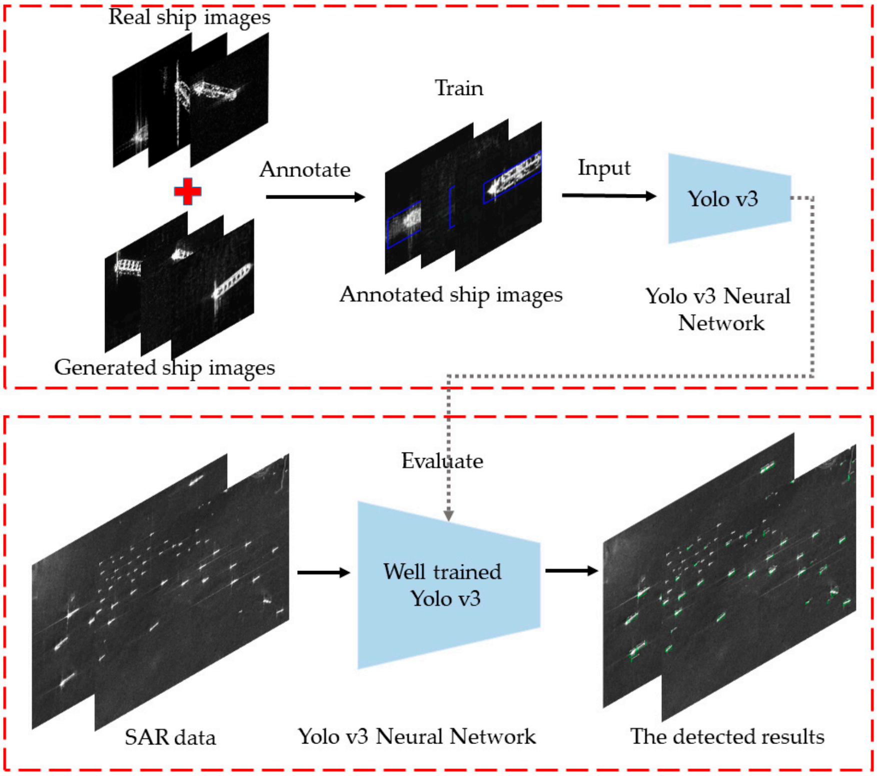


Figure 3. The flowchart of ship detection using Yolo v3 model and the composite dataset.

3. Experiment

3.1. Data Set Description

The Gaofen-3 (GF-3)3 m resolution SAR data are used in the experiment [54]. After obtaining the data, the radiometric calibration and normalization are carried out. Then, the "Imagelabel" tool is used to slice the original data; the process of making high-resolution SAR ship slices is shown in Figure 4. The slices include cargo ships, container ships, and tankers. The size of each slice is 256 × 256. The final number of slices is shown in Table 1.

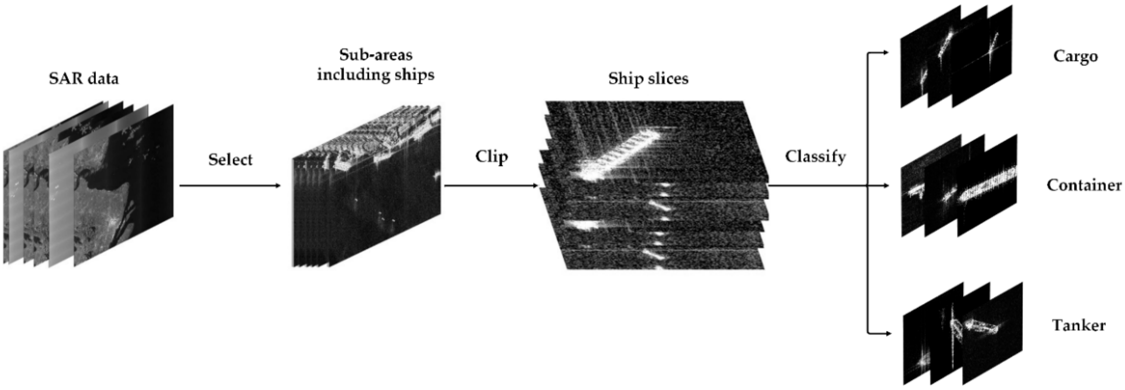


Figure 4. The process of making high-resolution Synthetic Aperture Radar (SAR) ship slices.

Table 1. Gaofen-3 (GF-3) high resolution ship slices.

Ship	Cargo	Container	Tanker	Total
Number	342	262	231	835

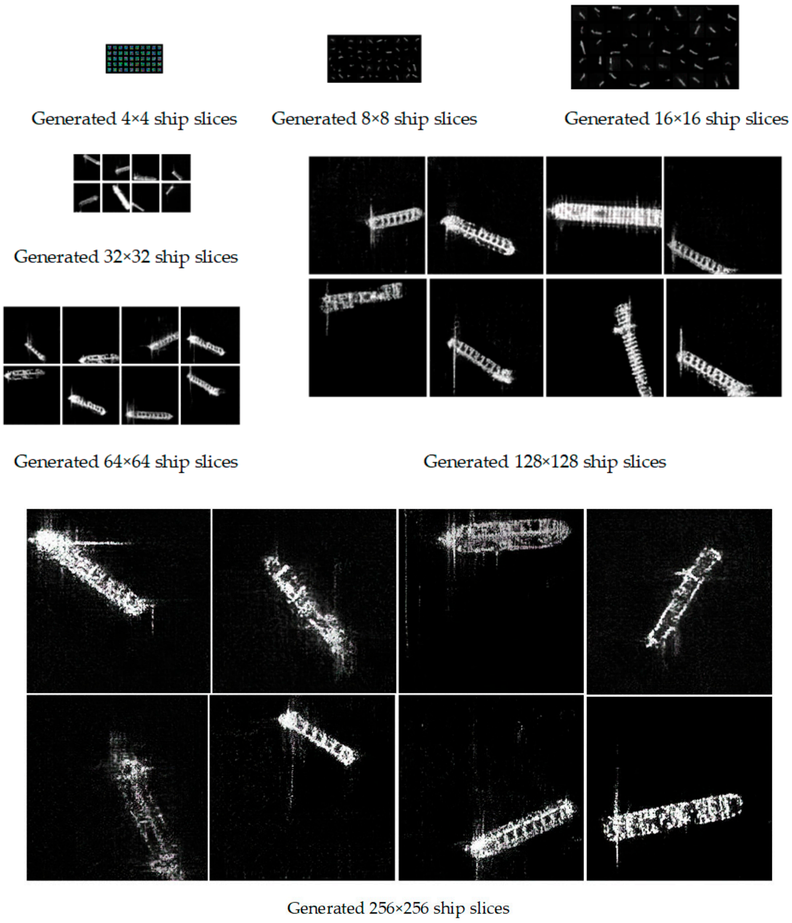
3.2. Network Parameters Setting

All experiments use the Pytorch deep learning framework (version 1.5.1). The Anaconda3 virtual environment is used for management. The Python version is 3.7, the CPU is an Intel i7-8700 K, and the GPU is an NVIDIA GeForce GTX 1080 Ti. The kernel size of the generator and discriminator filters in the convolutional layer is set to 3. The upsampling process is realized by nearest neighbor interpolation. The optimizer is "Adam", the learning rate is 0.001, and beta1 and beta2 are set to 0.9 and 0.999, respectively.

During Yolo v3 training, the batch size is set to 64, the learning rate is set to 0.001, and the Adam optimizer is used. The network input pixel is set to 256 × 256. The threshold of intersection over union (IOU) is set to 0.5, the thresholds for the target confidence range are set to 0.3 and 0.5, and the number of iterations is 10,000.

3.3. Experimental Results

Based on the real ship dataset, 4 × 4 to 256 × 256 high resolution SAR ship slices are generated by our proposed MW-ACGAN method, as shown in Figure 5. It can be seen from the results in the figure that the MW-ACGAN network has preliminarily realized the learning of three types of ship structures, which can simulate and characterize the unique image features of SAR images. The experimental results also show that in the process of generating 4 × 4 ~ 256 × 256 ship slices by MW-ACGAN, the image clarity and resolution are gradually improved, and the hull contour and texture are more realistic. It shows that with the increase in the number of network model layers, the image clarity is gradually improved, and the fitting ability of the generator to the real ship SAR slice is enhanced.



Generated 4×4 ship slices Generated 8×8 ship slices Generated 16×16 ship slices

Generated 32×32 ship slices

Generated 64×64 ship slices Generated 128×128 ship slices

Generated 256×256 ship slices

Figure 5. Different scales of SAR ship slices generated by the Multiscale Wasserstein Auxiliary Classifier Generative Adversarial Networks (MW-ACGAN).

Figure 6 shows ship slices generated by the ACGAN and the MW-ACGAN, respectively. Obviously, the two models can generate different kinds of ship slices and have good contour information. However, compared with MW-ACGAN model, ACGAN has obvious shortcomings. Some generated ship slices have certain fuzziness, and their texture features and edge features are also relatively fuzzy. The scattering sidelobe and speckle of the original ship slice are lost. In addition, due to the lack of network fitting ability, some generated ship slices appear with a curve shape, resulting in image distortion.

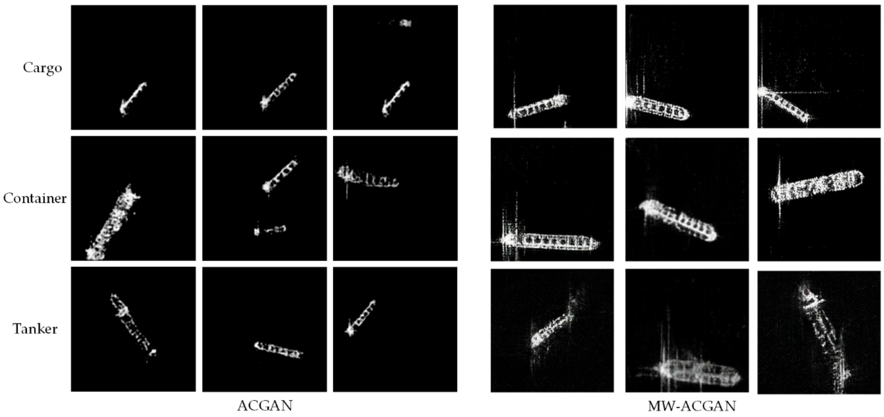


Figure 6. Comparison of ship slices generated by the ACGAN and the MW-ACGAN.

From Figures 5 and 6, we can see that our method can generate ship slices of different sizes and categories. In order to further analyze whether the ship slices generated by the MW-ACGAN have species diversity, we use the GAN test [55] method to evaluate he quality of the generated images. ResNet18 [56] is used as the benchmark classification network, as shown in Figure 7. Here, we change the last output layer of the original ResNet18 from 1000 categories to 3 categories to ensure consistency with the SAR ship data set. The image size is 256 × 256, the learning rate is 0.008, the batch size is 32, the Adam momentum optimizer is used, the betas are set to 0.5 and 0.99, and the loss function is CrossEntropyLoss. The final iteration epoch is set to 1000.

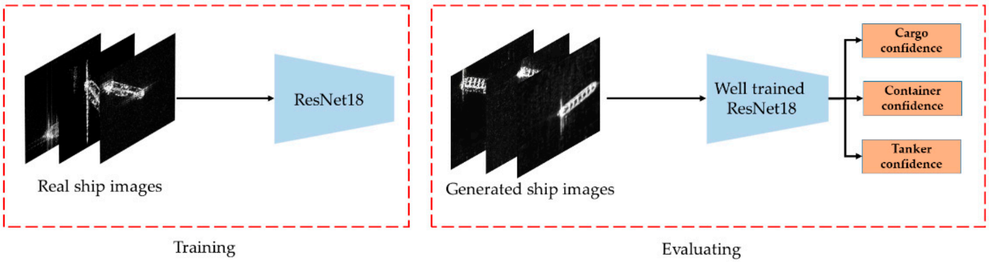


Figure 7. The Generative Adversarial Networks (GAN) test evaluation method based on ResNet18 network.

Firstly, the ResNet18 network is trained with the real 256 × 256 SAR ship data. The number of training set and test set is shown in Table 2. Then, the score of ResNet18 network is used as a quantitative evaluation method to generate SAR ship images. When the score level of the generated image is higher, the quality of the generated SAR image is higher, and the similarity with the real SAR image is greater.

Table 2. ResNet18 training set and test set.

Ship Types	S_{train}	S_{test}
Cargo	279	63
Container	211	51
Tanker	185	46
Total number	675	160

The ACGAN network and MW-ACGAN network are used to generate 1000 slices of cargo ships, container ships and oil tankers. Then, the slices are scored with the trained Rensnet18. Table 3 shows the ResNet18 scores for three different types of ships. The ResNet18 scores of MW-ACGAN in all ship categories are better than those of ACGAN, and the average score is better than 0.9. This shows that the method proposed in this paper can better understand the characteristics of ships and make ship slices more realistic.

Table 3. ResNet18 scores of the ACGAN and the MW-ACGAN.

Category	ACGAN's Average Score	MW-ACGAN's Average Score
Cargo	0.91	0.93
Container	0.78	0.88
Tanker	0.85	0.92
Average	0.85	0.91

To verify that the generated high-resolution ships are still practical for target detection scenarios, In order to verify the practicability of the high-resolution ship detection scene, Yolo v3 model is used to carry out ship detection experiments in SAR images.

Using the original ship set, the original ship set plus 2000 MW-ACGAN slices, and the original ship set plus 4000 MW-ACGAN slices as the training and test data, the Yolo v3 model is trained, respectively, to form three trained Yolo v3 models, and then the trained Yolo v3 models are tested on three real SAR images. In the whole training process, the number of iterations of different experiments remains unchanged.

Figure 8 and Table 4 show the detection results of three SAR images using the Yolo v3 model trained by the three sample sets, here the target confidence is set to 0.3. In Table 4, the Yolo v3 model after training in the original data set only detects two ships in the b SAR image. When 2000 and 4000 MW-ACGAN generated slices are added, 7 and 9 ships are detected. As expected, the use of the GAN to generate enhanced ship data can effectively increase the detection rate. The more images you add, the more the detection rate increases.

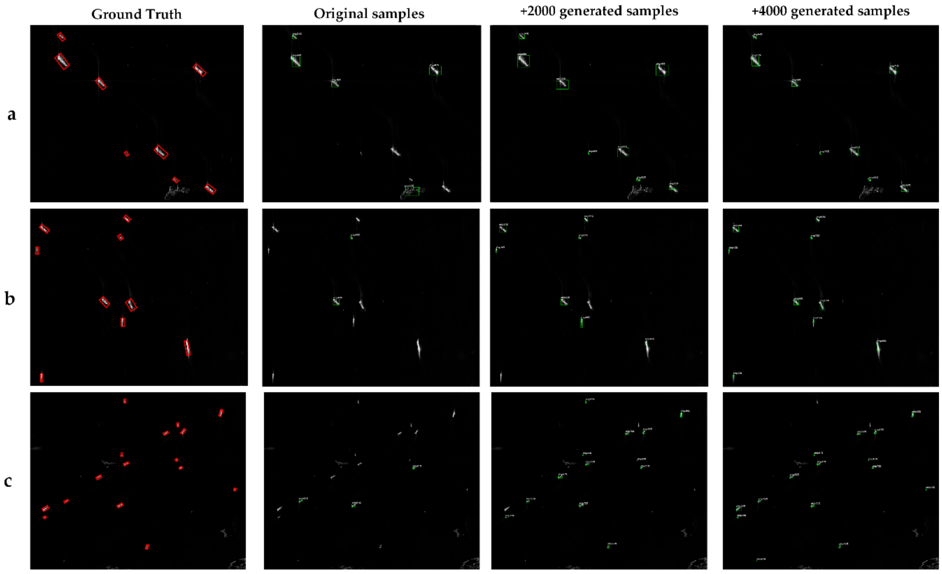


Figure 8. Detection results of (**a**,**b**,**c**) SAR images using Yolo v3 model trained by different sample sets.

Table 5 shows the statistics of Yolo v3 detection results. Using only the original training data, the detection rate of Yolo v3 is only 27% when the target confidence is 0.3. This may be due to the fact that the original samples are too small to overfit the model after 10,000 iterations. When 2000 generated slices are added, the detection rate is increased by 58%, which indicates that the number of ship modes in the data set is increased and the generalization ability of the model is improved. The detection rate reaches 94% when 4000 slices are added to the original data set, which indicates that the ship image generated by MW-ACGAN has high quality and multiclass features, which can effectively improve the generalization ability of the detection model and play an important role in the detection scenarios.

Table 4. Statistics of Yolo v3 detection results.

Image ID	Object Confidence >= 0.3							Object Confidence >= 0.5					
	Total Number of Ships	835 Slices		+2000 Slices		+4000 Slices		835 Slices		+2000 Slices		+4000 Slices	
		Detec-Ted	False Alarm	Detec-Ted	False Alarm	Detec-Ted	False Alarm	Detec-Ted	False Alarm	Detec-Ted	False Alarm	Detec-Ted	False Alarm
a	8	4	1	8	0	8	0	2	0	5	0	8	0
b	9	2	0	7	0	9	0	2	0	3	0	7	0
c	16	3	0	13	0	15	0	2	0	8	0	8	0

Table 5. Statistics of Yolo v3 detection results.

	Object Confidence ≥ 0.3		Object Confidence ≥ 0.5	
	Accuracy	False Alarm Rate	Accuracy	False Alarm Rate
835 real ship slices	27%	3%	18%	0
+2000 MW-ACGAN generated slices	85%	0	48%	0
+4000 MW-ACGAN generated slices	94%	0	70	0

4. Conclusions

In order to meet the application requirements of ship detection in high-resolution SAR images with small samples, we propose a ship detection method based on the SAR image generation network (MW-ACGAN) and the detection network Yolo v3 under small samples. The experimental results of GF-3 SAR data show that the MW-ACGAN network can effectively generate realistic multiclass ship images by using the Wasserstein distance and gradient penalty. The multi-scale loss term can effectively generate multi-scale ship images. The generated ship slices show good class separability under ResNet18 classifier, which can also be applied to the study of ship classification in the future. In the process of Yolo v3 network detection, the original small sample SAR data set is effectively expanded by using the generated samples, and the detection accuracy of three SAR images is better than 90%, which proves the feasibility of our method.

In the future, we will further study the generation of SAR ship images under different sea conditions, different polarization modes, and different incident angles. We will combine deep learning abstract features and traditional concrete ones to further improve the accuracy of the generated features. In the next step, more systematic experimental analysis will be carried out, and generated images will be applied to ship detection and classification under multiple scenes and complex sea conditions, so as to improve the detection rate of ships in complex scenarios.

Author Contributions: Conceptualization, L.Z. and H.Z.; methodology, L.Z. and H.Z.; software, L.Z. and C.W.; validation, L.Z. and F.G.; formal analysis, L.Z. and F.G.; investigation, H.Z. and F.W.; resources, H.Z. and F.W.; data curation, L.Z. and H.Z.; writing—original draft preparation, L.Z. and H.Z.; visualization, L.Z. and F.W.; supervision, H.Z. and C.W.; writing—review and editing, H.Z.; project administration, H.Z. and C.W.; funding acquisition C.W. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Natural Science Foundation of China under Grant 41930110.

Acknowledgments: The authors would like to express their gratitude to the China Center for Resources Satellite Data and Application for providing Gaofen-3 images.

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

References

1. De Maio, A.; Orlando, D.; Pallotta, L.; Clemente, C. A Multifamily GLRT for Oil Spill Detection. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 63–79. [CrossRef]
2. Lombardo, P.; Oliver, C.J. Optimum detection and segmentation of oil slicks using polarimetric SAR data. In Proceedings of the 2000 IEEE International Conference on Radar, Alexandria, VA, USA, 12 May 2000; pp. 309–321.
3. Kanjir, U.; Greidanus, H.; Oštir, K. Vessel detection and classification from spaceborne optical images: A literature survey. *Remote Sens. Environ.* **2018**, *207*, 1–26. [CrossRef]
4. Lombardo, P.; Sciotti, M. Segmentation-based technique for ship detection in SAR images. *IEE Proc. Radar Sonar Navig.* **2001**, *148*, 147–159. [CrossRef]
5. Protopapadakis, E.; Voulodimos, A.; Doulamis, A.; Doulamis, N.; Dres, D.; Bimpas, M. Stacked autoencoders for outlier detection in over-the-horizon radar signals. *Comput. Intell. Neurosci.* **2017**, *2017*, 5891417. [CrossRef] [PubMed]
6. Zhang, Y.; Li, Q.-Z.; Zhang, F.-N. Ship detection for visual maritime surveillance from non-stationary platforms. *Ocean Eng.* **2017**, *141*, 53–63. [CrossRef]
7. Makantasis, K.; Protopapadakis, E.; Doulamis, A.; Matsatsinis, N. Semi-supervised vision-based maritime surveillance system using fused visual attention maps. *Multimed. Tools Appl.* **2016**, *75*, 15051–15078. [CrossRef]
8. Tello, M.; López-Martínez, C.; 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]
9. Touzi, R.; Charbonneau, F.J.; Hawkins, R.K.; Vachon, P.W. Ship detection and characterization using polarimetric SAR. *Can. J. Remote Sens.* **2004**, *30*, 552–559. [CrossRef]
10. Rostami, M.; Kolouri, S.; Eaton, E.; Kim, K. Deep Transfer Learning for Few-shot SAR Image Classification. *Remote Sens.* **2019**, *11*, 1374. [CrossRef]

11. 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]
12. Liu, L.; Chen, G.; Pan, Z.; Lei, B.; An, Q. Inshore Ship Detection in Sar Images Based on Deep Neural Networks. In Proceedings of the 2018 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), Valencia, Spain, 22–27 July 2018; pp. 25–28.
13. Chen, S.; Zhang, J.; Zhan, R. R2FA-Det: Delving into high-quality rotatable boxes for ship detection in SAR images. *Remote Sens.* **2020**, *12*, 2031. [CrossRef]
14. Cui, Z.; Wang, X.; Liu, N.; Cao, Z.; Yang, J. Ship detection in large-scale SAR images via spatial shuffle-group enhance attention. *IEEE Trans. Geosci. Remote Sens.* **2020**, 1–13. [CrossRef]
15. Zhao, Y.; Zhao, L.; Xiong, B.; Kuang, G. Attention receptive pyramid network for ship detection in SAR images. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2020**, *13*, 2738–2756. [CrossRef]
16. Dong, Y.; Wang, C.; Zhang, H.; Wang, Y.; Zhang, B. Impact Analysis of Incident Angle Factor on High-Resolution Sar Image Ship Classification Based on Deep Learning. In Proceedings of the 2019 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), Yokohama, Japan, 28 July–2 August 2019; pp. 1358–1361.
17. 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]
18. Lin, Z.; Ji, K.; Leng, X.; Kuang, G. Squeeze and excitation rank Faster R-CNN for ship detection in SAR images. *IEEE Geosci. Remote Sens. Lett.* **2019**, *16*, 751–755. [CrossRef]
19. Deng, Z.; Sun, H.; Zhou, S.; Zhao, J.; Lei, L.; Zou, H. Multi-scale object detection in remote sensing imagery with convolutional neural networks. *ISPRS J. Photogramm. Remote Sens.* **2018**, *145*, 3–22. [CrossRef]
20. Deng, Z.; Sun, H.; Zhou, S.; Zhao, J. Learning deep ship detector in SAR images from scratch. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 4021–4039. [CrossRef]
21. Zhao, J.; Guo, W.; Zhang, Z.; Yu, W. A coupled convolutional neural network for small and densely clustered ship detection in SAR images. *Sci. China Inf. Sci.* **2018**, *62*, 42301. [CrossRef]
22. Zhao, J.; Zhang, Z.; Yu, W.; Truong, T.-K. A cascade coupled convolutional neural network guided visual attention method for ship detection from SAR images. *IEEE Access.* **2018**, *6*, 50693–50708. [CrossRef]
23. Wang, C.; Zhang, H.; Wu, F.; Zhang, B.; Tian, S. Ship classification with deep learning using COSMO-SkyMed SAR data. In Proceedings of the 2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), Fort Worth, TX, USA, 23–28 July 2017; pp. 558–561.
24. Li, J.; Qu, C.; Shao, J. Ship detection in SAR images based on an improved Faster R-CNN. In Proceedings of the 2017 SAR in Big Data Era: Models, Methods and Applications (BIGSARDATA), Beijing, China, 13–14 November 2017; pp. 1–6.
25. Sun, X.; Wang, Z.; Sun, Y.; Diao, W.; Zhang, Y.; Fu, K. AIR-SARShip-1.0: High-resolution SAR ship detection dataset. *J. Radars* **2019**, *8*, 852–862.
26. 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]
27. Ren, Z.; Hou, B.; Wu, Q.; Wen, Z.; Jiao, L. A Distribution and Structure Match Generative Adversarial Network for SAR Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2020**, *58*, 3864–3880. [CrossRef]
28. Ulaby, F.T.; Moore, R.K.; Fung, A.K. *Microwave Remote Sensing: Active and Passive. Volume 3-from Theory to Applications*; Artech House Press: Norwood, MA, USA, 1986.
29. Franceschetti, G.; Iodice, A.; Riccio, D.; Ruello, G. SAR raw signal simulation for urban structures. *IEEE Trans. Geosci. Remote Sens.* **2003**, *41*, 1986–1995. [CrossRef]
30. Xu, F.; Jin, Y.-Q. Imaging simulation of polarimetric SAR for a comprehensive terrain scene using the mapping and projection algorithm. *IEEE Trans. Geosci. Remote Sens.* **2006**, *44*, 3219–3234. [CrossRef]
31. Brunner, D.; Lemoine, G.; Greidanus, H.; Bruzzone, L. Radar imaging simulation for urban structures. *IEEE Geosci. Remote Sens. Lett.* **2010**, *8*, 68–72. [CrossRef]
32. Balz, T. Real-time SAR simulation on graphics processing units. In Proceedings of the 2006 European Conference on Synthetic Aperture Radar (EUSAR), Dresden, Germany, 16–18 May 2006; pp. 1–4.
33. Auer, S.; Bamler, R.; Reinartz, P. RaySAR-3D SAR simulator: Now open source. In Proceedings of the 2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), Beijing, China, 10–15 July 2016; pp. 6730–6733.

34. Hammer, H.; Kuny, S.; Schulz, K. Amazing SAR imaging effects-explained by SAR simulation. In Proceedings of the 2014 European Conference on Synthetic Aperture Radar (EUSAR), Berlin, Germany, 3–6 June 2014; pp. 1–4.
35. Niu, S.; Qiu, X.; Lei, B.; Ding, C.; Fu, K. Parameter Extraction Based on Deep Neural Network for SAR Target Simulation. *IEEE Trans. Geosci. Remote Sens.* **2020**, *58*, 4901–4914. [CrossRef]
36. Karras, T.; Laine, S.; Aila, T. A style-based generator architecture for generative adversarial networks. In Proceedings of the 2019 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–20 June 2019; pp. 4401–4410.
37. Brock, A.; Donahue, J.; Simonyan, K. Large Scale GAN Training for High Fidelity Natural Image Synthesis. *arXiv* **2020**, arXiv:1809.11096.
38. Bissoto, A.; Perez, F.; Valle, E.; Avila, S. Skin Lesion Synthesis with Generative Adversarial Networks. In Proceedings of the 2018 ISIC Skin Image Analysis Workshop and Challenge, Granada, Spain, 20 September 2018; pp. 294–302.
39. Park, T.; Liu, M.-Y.; Wang, T.-C.; Zhu, J.-Y. Semantic Image Synthesis with Spatially-Adaptive Normalization. In Proceedings of the 2019 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–20 June 2019; pp. 2337–2346.
40. Ledig, C.; Theis, L.; Huszar, F.; Caballero, J.; Cunningham, A.; Acosta, A.; Aitken, A.; Tejani, A.; Totz, J.; Wang, Z.; et al. Photo-realistic single image super-resolution using a generative adversarial network. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4681–4690.
41. Wang, X.; Yu, K.; Wu, S.; Gu, J.; Liu, Y.; Dong, C.; Qiao, Y.; Change Loy, C. Esrgan: Enhanced super-resolution generative adversarial networks. In Proceedings of the 2018 European Conference on Computer Vision, Munich, Germany, 8–14 September 2018; pp. 63–79.
42. Odena, A.; Olah, C.; Shlens, J. Conditional image synthesis with auxiliary classifier GANs. In Proceedings of the 2017 International conference on machine learning (ICML), Sydney, Australia, 6–11 August 2017; pp. 2642–2651.
43. Guo, J.; Lei, B.; Ding, C.; Zhang, Y. Synthetic Aperture Radar Image Synthesis by Using Generative Adversarial Nets. *IEEE Geosci. Remote Sens. Lett.* **2017**, *14*, 1111–1115. [CrossRef]
44. Schwegmann, C.P.; Kleynhans, W.; Salmon, B.P.; Mdakane, L.W.; Meyer, R.G.V. Synthetic aperture radar ship discrimination, generation and latent variable extraction using information maximizing generative adversarial networks. In Proceedings of the 2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), Fort Worth, TX, USA, 23–28 July 2017; pp. 2263–2266.
45. Marmanis, D.; Yao, W.; Adam, F.; Datcu, M.; Reinartz, P.; Schindler, K.; Wegner, J.D.; Stilla, U. Artificial Generation of Big Data for Improving Image Classification: A Generative Adversarial Network Approach on SAR Data. *arXiv* **2017**, arXiv:1711.02010.
46. Goodfellow, I.J.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative Adversarial Networks. *arXiv* **2020**, arXiv:1406.2661. [CrossRef]
47. Gulrajani, I.; Ahmed, F.; Arjovsky, M.; Dumoulin, V.; Courville, A.C. Improved training of wasserstein GANs. In Proceedings of the 2017 Advances in neural information processing systems, Long Beach, CA, USA, 4–9 December 2017; pp. 5767–5777.
48. Redmon, J.; Farhadi, A. YOLO v3: An Incremental Improvement. *arXiv* **2020**, arXiv:1804.02767.
49. Zhang, T.; Zhang, X.; Shi, J.; Wei, S. High-speed ship detection in SAR images by improved yolov3. In Proceedings of the 2019 16th International Computer Conference on Wavelet Active Media Technology and Information Processing, Chengdu, China, 4–15 December 2019; pp. 149–152.
50. Gao, F.; He, Y.; Wang, J.; Hussain, A.; Zhou, H. Anchor-free Convolutional Network with Dense Attention Feature Aggregation for Ship Detection in SAR Images. *Remote Sens.* **2020**, *12*, 2619. [CrossRef]
51. Zhang, T.; Zhang, X. ShipDeNet-20: An Only 20 Convolution Layers and <1-MB Lightweight SAR Ship Detector. *IEEE Geosci. Remote Sens. Lett.* **2020**, 1–5. [CrossRef]
52. Chen, X.; Qi, L.; Yang, Y.; Postolache, O.; Yu, Z.; Xu, X. Port ship detection in complex environments. In Proceedings of the 2019 International Conference on Sensing and Instrumentation in IoT Era (ISSI), Lisbon, Portugal, 29–30 August 2019; pp. 1–6.

53. Wang, Z.; Zhou, Y. A HED-optimized Automatic Detection and Tracking Algorithm for Marine Moving Targets based on YOLO V3. *J. Phys. Conf. Ser.* **2020**, *1449*, 012126. [CrossRef]
54. Dong, Y.; Zhang, H.; Wang, C.; Wang, Y. Fine-grained ship classification based on deep residual learning for high-resolution SAR images. *Remote Sens. Lett.* **2019**, *10*, 1095–1104. [CrossRef]
55. Bissoto, A.; Valle, E.; Avila, S. The Six Fronts of the Generative Adversarial Networks. *arXiv* **2019**, arXiv:1910.13076.
56. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 25 June–1 July 2016; pp. 770–778.

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