



Article

Lightweight CFARNets for Landmine Detection in Ultrawideband SAR

Yansong Zhang, Yongping Song and Tian Jin *

College of Electronic Science and Technology, National University of Defense Technology,
Changsha 410073, China; zhangyansong18@nudt.edu.cn (Y.Z.); songyongpin08@nudt.edu.cn (Y.S.)

* Correspondence: tianjin@nudt.edu.cn

Abstract: The high-resolution image obtained by ultrawideband synthetic aperture radar (UWB SAR) includes rich features such as shape and scattering features, which can be utilized for landmine discrimination and detection. Due to the high performance and automatic feature learning ability, deep network-based detection methods have been widely employed in SAR target detection. However, existing deep networks do not consider the target characteristics in SAR images, and their structures are too complicated. Therefore, lightweight deep networks with efficient and interpretable blocks are essential. This work investigates how to utilize the SAR characteristics to design a lightweight deep network. The widely employed constant false alarm rates (CFAR) detector is used as a prototype and transformed into trainable multiple-feature network filters. Based on CFAR filters, we propose a new class of networks called CFARNets which can serve as an alternative to convolutional neural networks (CNNs). Furthermore, a two-stage detection method based on CFARNets is proposed. Compared to prevailing CNNs, the complexity and number of parameters of CFARNets are significantly reduced. The features extracted by CFARNets are interpretable as CFAR filters have definite physical significance. Experimental results show that the proposed CFARNets have comparable detection performance compared to other real-time state-of-the-art detectors but with faster inference speed.

Keywords: synthetic aperture radar (SAR); landmine; CFAR network; target detection; lightweight network



Citation: Zhang, Y.; Song, Y.; Jin, T.
Lightweight CFARNets for Landmine
Detection in Ultrawideband SAR.
Remote Sens. **2023**, *15*, 4411. <https://doi.org/10.3390/rs15184411>

Academic Editor: Andrzej Stateczny

Received: 29 July 2023

Revised: 28 August 2023

Accepted: 5 September 2023

Published: 7 September 2023



Copyright: © 2023 by the authors.
Licensee MDPI, Basel, Switzerland.
This article is an open access article
distributed under the terms and
conditions of the Creative Commons
Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

There are many landmines distributed around the world. Landmines have strong concealment and destructive power, greatly reducing the mobility of troops and interfering with civilian activity. Therefore, efficiently and quickly detecting and eliminating landmines is of great significance for both military and people's livelihoods. To detect landmines, sensing sensors such as ground-penetrating radars (GPR) which can penetrate the soil are essential [1,2]. Synthetic aperture Radar (SAR) is another powerful sensor for target detection, which is capable of obtaining a high-resolution image of the targets and their surrounding areas. With the development of SAR techniques including hardware, waveform design, and imaging algorithm, the resolution and imaging quality have been greatly improved in past decades [3–6]. The ultrawideband synthetic aperture radar (UWB SAR) has become an important means for detecting landmines. Although the UWB SAR image can reach a resolution of centimeters, landmines in the UWB SAR image are embedded in complicated environments with small sizes, weak usable features, and low SNR [3,4]. These factors make landmine detection difficult.

The detection of landmines in UWB SAR images shares many merits with the detection of other targets in SAR images, such as vehicles and ships. For example, the processing steps of landmines and other targets are similar [3,4]. The three processing stages proposed by the MIT Lincoln Laboratory, which include detection, discrimination, and classification, can apply to landmines as well [3,4,7,8]. Numerous methods have been proposed for landmine

detection in SAR images, which can be classified into traditional detection methods and deep learning-based methods.

1.1. Traditional Detection Methods

One of the popular detection methods is the constant false alarm rates (CFAR) detector [3,4,9], which is mainly based on the amplitude divergence between the target and the clutter. The CFAR detector can be viewed as a single-feature method. Although the performance of the CFAR detector is limited in complex clutter scenes, it has advantages in two aspects: the computation speed is fast and it is simple to implement. As a result, the research of CFAR detectors is still active currently, and several invariants of CFAR detectors are proposed. One major improvement is to extend the traditional pixel-level CFAR to the superpixel-level CFAR [10–12], which can retain the complete information of a target and reduce false alarms. The second improvement is to combine CFAR detectors with deep learning networks [13,14]. In [13], a CFAR detector is used as a preprocessing stage of deep networks. In [14], the CFAR indicative maps are used to guide the classification loss function and feature extraction in deep networks.

As the SAR images contain rich target features such as shape and scattering features, features have played an important role in target detection. In addition to the amplitude feature, hand-crafted features are used for landmine detection as well. In [3], the double hump signature of the landmine in the UWB SAR image was analyzed and extracted, which was then used for landmine detection. In [4], the scale-invariant feature transformation (SIFT) was applied to the UWB SAR image to extract the landmine's features, and then feature point matching (FPM) was carried out based on the reference image to test whether the region of interest (ROI) is a landmine or clutter. In [15], the histogram of degree (HOG) features was used for landmine detection by GPR radar. These features-based methods have been proven to be useful for landmine detection. However, the hand-crafted features are susceptible to target and environment variation, leading to performance degradation in complex scenes.

1.2. Deep Learning-Based Methods

To overcome the problem of hand-crafted features, researchers have resorted to deep networks that can learn the high-level features from the images automatically. Convolutional neural networks (CNNs) have been widely used for SAR target classification due to their high performance and efficient training procedure [7,16,17]. To reduce the overfitting problem, Chen et al. [7] proposed all-convolutional networks (A-ConvNets) without fully connecting layers. Kwak et al. [16] investigated the effect of speckle noise on CNN features and introduced regularization in the training procedure to minimize the noise effect. Wang et al. [17] proposed a CNN-based feature fusion method for target discrimination that jointly uses intensity and edge information of SAR images. Technically speaking, deep networks proposed for target classification have no target locating ability and should be combined with external target proposal generation methods such as CFAR or sliding window. Target proposal generation can be achieved by network modules as well, i.e., networks can be used to jointly locate and identify the targets. Representative methods include SSD, Faster-RCNN, YOLO, and other deep network-based detectors [18,19]. With the development of deep learning techniques, novel network modules and learning paradigms have been introduced in the field of SAR as well, such as attention mechanism [20,21], transfer learning [22], semi-supervised learning [23], etc. It should be pointed out that the above deep networks are designed for other targets and there is currently no deep learning-based detection method customized for landmines. By carefully choosing the parameters, these methods can be applied to landmine detection with a high performance. However, the above deep learning-based methods do not consider the specialty of SAR images, and the employed networks are too sophisticated in terms of both network architectures and parameters. Furthermore, the behavior and extracted features of these methods are difficult to explain.

1.3. Contribution of This Work

The deficiency of deep networks in SAR target detection encourages the authors to develop novel networks that can utilize the SAR characteristics for landmine detection. We propose a new class of filter, block, and networks that can serve as an alternative to that in convolutional neural networks (CNNs) for landmine detection in SAR images. Inspired by the recent development of the CFAR technique which is implemented via GPU tensor operation [9], the authors re-investigated the possibility of incorporating CFAR in deep networks. It was found that CFAR detector can be implemented by using tensor operations such as convolution and mean pooling, which not only accelerates the computation speed but also makes it possible to be incorporated into the network as an automatic differentiation operation. Based on this, different CFAR blocks and deep CFAR networks which are coined as CFARNets are proposed for landmine detection. The main contributions of this work are as follows:

- (1) It is found that cell averaging-CFAR (CA-CFAR) can be implemented via mean pooling, which not only accelerates the computation speed but also makes it possible to be used as an interpretable network module. Based on this finding, we propose a CFAR filter, a new class of filter that can capture the divergence of SAR images.
- (2) We propose CFAR blocks, consisting of CFAR filters and other nonlinear filters, which can replace standard convolutional layers. We also propose lightweight CFARNets based on the developed CFAR blocks, which have low complexity and few parameters.
- (3) A two-stage landmine detection method based on CFARNets is proposed. Since CFARNets are developed based on the CFAR filter which has definite physical significance, they can efficiently utilize the SAR characteristics and are interpretable.
- (4) Experiments on landmines are carried out. The performance of the proposed detection method based on CFARNet is comparable to YOLO detectors but with higher inference speed.

The rest of this work is organized according to the pipeline of developing novel deep learning-based methods, which follows the filter-block-network-detector order. First, the background and implementation of the CFAR filter are presented in Section 3. Following that, details of the proposed CFAR blocks and deep CFAR networks for target discrimination are provided in Section 3. Section 4 illustrates the two-stage target detection method based on deep CFAR networks and Section 5 provides the experimental results.

2. CFAR Filter Based on CA-CFAR

Filters are the basic module of deep networks. The deep networks are trained by back-propagation. Most deep learning frameworks such as Tensorflow and Pytorch use automatic differentiation operations to construct the networks. The newly developed filters should be composed of their support tensor operations. In this work, we would like to use the most widely detection method, i.e., the CFAR detector, to construct deep network. Thus, the bridge between the CFAR detector and common tensor operations should be built. In particular, the CA-CFAR detector is used as a prototype and transformed into a trainable network filter.

2.1. Review of CA-CFAR

CA-CFAR is widely used for its computational efficiency and easy implementation. The CA-CFAR algorithm uses the samples in the sliding window to estimate the clutter level. The test rule and threshold are dependent on the clutter model. For simplicity, assume that the clutter follows an exponential distribution. Denote $z(i, j)$ as the cell under test (CUT), the following rule is used to determine whether it is a target pixel [9]

$$z(i, j) \underset{H_0}{\overset{H_1}{\gtrless}} \alpha \bar{Z}(i, j), \quad (1)$$

where H_0 and H_1 stand for the absence and presence of a target pixel, respectively. $\alpha = \ln(P_{FA})$ is the multiplier controlling the probability of false alarms. $\bar{Z}(i, j)$ is the mean amplitude of the clutter in the sliding window, which is given by

$$\bar{Z}(i, j) = \frac{1}{N_s} \sum_{\substack{\frac{g-1}{2} < |n| \leq \frac{k-1}{2} \\ \frac{g-1}{2} < |m| \leq \frac{k-1}{2}}} z(i-m, j-n), \quad (2)$$

where k and g are the sizes of the square sliding window and guard area (For simplicity, k and g are odd numbers), respectively, and $N_s = k^2 - g^2$ is the number of used samples in the sliding window. Equation (2) is referred to as CA operation. A demonstration of the sliding window mode in CA-CFAR is shown in Figure 1.

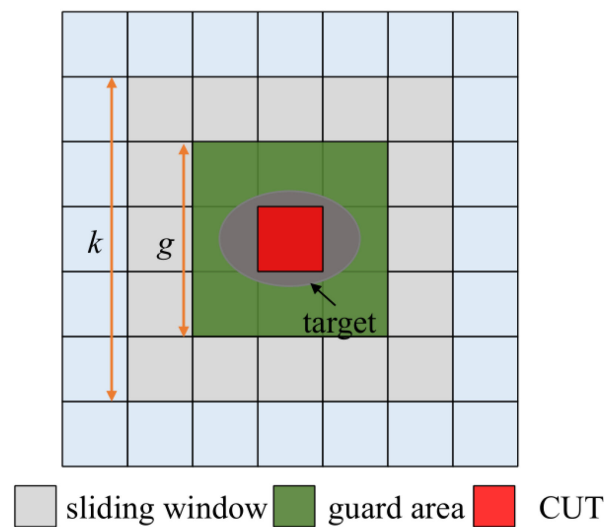


Figure 1. Demonstration of the sliding window mode in CA-CFAR.

2.2. CFAR Filter

To speed up CFAR detectors, Ref. [9] suggested that the CFAR detectors can be implemented via graphics processing unit (GPU) tensor operations, including tensor convolution, shift, and Boolean operation. Inspired by this idea, we would like to use the tensor operation for CFAR implementation which not only accelerates the computation speed but also makes it possible to be incorporated into the network as an automatic differentiation operation.

Equation (2) can be transformed into different equivalent formulas, the first one is

$$\begin{aligned} \bar{Z}(i, j) &= \frac{1}{N_s} \sum_{\substack{|n| \leq \frac{k-1}{2} \\ |m| \leq \frac{k-1}{2}}} z(i-m, j-n) - \frac{1}{N_s} \sum_{\substack{|n| \leq \frac{g-1}{2} \\ |m| \leq \frac{g-1}{2}}} z(i-m, j-n) \\ &= \sum_{\substack{|n| \leq \frac{k-1}{2} \\ |m| \leq \frac{k-1}{2}}} z(i-m, j-n) \mathbf{W}_k(m, n) - \sum_{\substack{|n| \leq \frac{g-1}{2} \\ |m| \leq \frac{g-1}{2}}} z(i-m, j-n) \mathbf{W}_g(m, n) \end{aligned} \quad (3)$$

where $\mathbf{W}_k = 1/N_s \cdot \mathbf{1}_{k \times k}$ and $\mathbf{W}_g = 1/N_s \cdot \mathbf{1}_{g \times g}$ are the kernel matrixes of the sliding window and the guard area, respectively. Equation (3) can be implemented by convolution operation [9].

Another equivalent formula for Equation (2) is

$$\bar{Z}(i, j) = \frac{k^2}{N_s} \cdot \frac{1}{k^2} \sum_{\substack{|n| \leq \frac{k-1}{2} \\ |m| \leq \frac{k-1}{2}}} z(i-m, j-n) - \frac{g^2}{N_s} \cdot \frac{1}{g^2} \sum_{\substack{|n| \leq \frac{g-1}{2} \\ |m| \leq \frac{g-1}{2}}} z(i-m, j-n), \quad (4)$$

The two terms in (4) can be computed by averaging on a defined window followed by a multiplier, where the averaging on a defined window can be implemented by the mean pooling operation. The kernel size of the mean pooling for the first term is k , while the kernel size for the second term is g . Note that it is essential to align the values of the two terms in (3) and (4), and the input image should be padded with $\frac{k-1}{2}$ and $\frac{g-1}{2}$ zeros on each side before applying convolution and mean pooling, respectively. Figure 2 shows the systematic view of these two implementations.

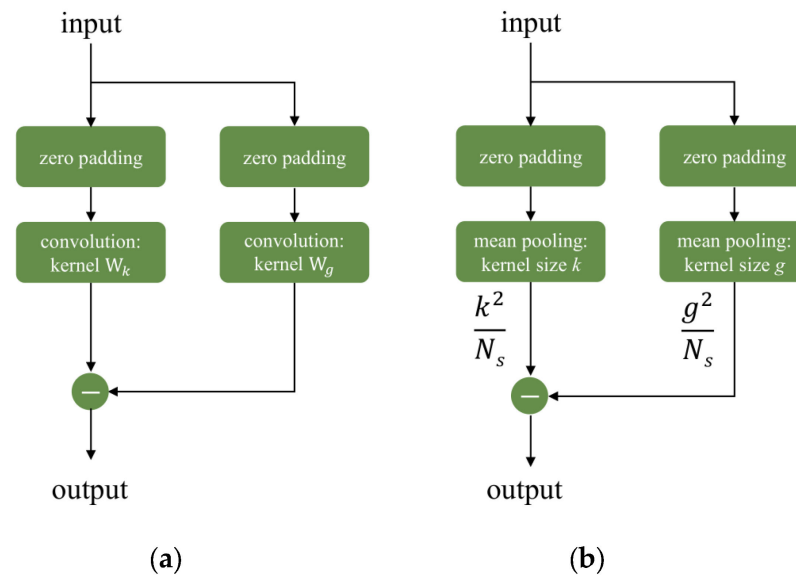


Figure 2. A systematic view of the CA operation. (a) Convolution-based implementation; (b) mean pooling-based implementation.

The mean pooling-based implementation merely requires addition, which is cheaper than the convolution-based implementation. It is adopted to implement the CA Operation and used as a basic filter of deep networks. The mean pooling is a standard operation in deep learning frameworks such as Pytorch, Tensorflow, etc. Thus, the mean pooling-based implementation enables the network to incorporate CFAR as an automatic differentiation operation in deep networks.

Equation (1) can be rewritten as

$$z(i, j) - \alpha \bar{Z}(i, j) \begin{matrix} \geq 0, \\ \leq 0, \end{matrix} \begin{matrix} H_1 \\ H_0 \end{matrix} \quad (5)$$

In the above equation, the left term $z(i, j) - \alpha \bar{Z}(i, j)$ is the amplitude divergence between the CUT and clutter level, while the residual term is just a threshold decision that can be replaced by other sophisticated operation in the network. Thus, $z(i, j) - \alpha \bar{Z}(i, j)$ is considered a basic filter for networks, which is referred to as CFAR filter in the following. Figure 3 shows the computation process of $z(i, j) - \alpha \bar{Z}(i, j)$, where α is a trainable parameter. To distinguish from the convolution block, the sizes of the sliding window and the guard area in the CA operation are denoted as $k \square g$, as $\square$ can represent the processing procedure of CFAR which includes two windows.

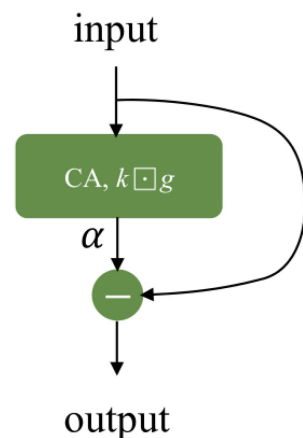


Figure 3. A systematic view of the CFAR filter.

For deep networks, a tensor $F \in \mathbb{R}^{M \times N \times C}$ is usually used to represent the extracted feature map. If we apply the CFAR filter to each channel separately, the p th channel of the output feature map can be written as

$$\hat{F}_{:,p} = F_{:,p} - \alpha \bar{F}_{:,p}, \quad (6)$$

where $0 \leq p \leq C - 1$, $\bar{F}_{:,p} \in \mathbb{R}^{M \times N}$ is the sliding mean value of $F_{:,p} \in \mathbb{R}^{M \times N}$ estimated by CA (Index (i, j) in Equation (5) is omitted). For simplicity, Equation (6) is also denoted as $\hat{F} = f_{k \odot g}^{CFAR}(F)$.

3. CFAR Blocks and Deep CFAR Networks

By combining the widely used network architecture and filters, one can design blocks and deep networks for target recognition and detection based on the CFAR filter. Two key works are essential to achieve this. First, network blocks with multiple feature extraction ability should be designed based on the single-feature CFAR filter. Second, other nonlinear filters should be added to improve the nonlinearity of network blocks. Deep CFAR networks are composed of several CFAR blocks and classifiers, which are trained based on clutter patches and target patches. After training, deep networks can be used to discriminate the clutter and target.

3.1. CFAR Blocks

Deep networks can be viewed as a combination of multiple nonlinear functions. The nonlinearity is an important factor in why deep networks work. However, the CFAR filter $z(i, j) - \alpha \bar{z}(i, j)$ is mainly based on sample averaging and subtraction, which is a linear function. To improve the representation ability of the CFAR filter, it is combined with other non-linear operations. A single-branch CFAR block was first designed based on the CFAR filter and 1×1 convolution [24]. Following that, two invariants were proposed according to the prevailing micro-architectures in deep networks.

(1) Single-branch CFAR (S-CFAR) Block

The incorporation of 1×1 convolution and rectification function can increase the nonlinearity of the network with low computational cost [24]. It is a common choice in the design of the network. Given this, S-CFAR block is designed by stacking two 1×1 convolution layers and the CFAR filter together, which is shown in Figure 4a. The 1×1 convolution layers are followed by batch normalization and rectified linear unit (ReLU) activation function. In the S-CFAR block, we used the same number of channels between 1×1 convolution layers and the output features, which is denoted by C_0 .

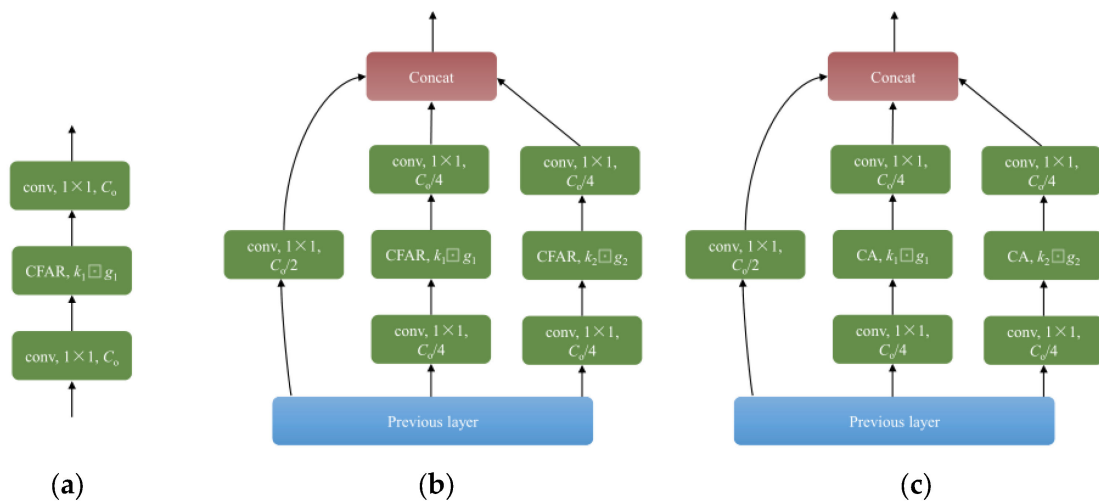


Figure 4. CFAR blocks. (a) Single branch CFAR Block; (b) Inception CFAR block I; (c) Inception CFAR block II.

Denote the input of the S-CFAR block by tensor $\mathbf{X} \in \mathbb{R}^{M \times N \times C}$, kernels of the first and second 1×1 convolution layers by tensor $\mathbf{W}^1 \in \mathbb{R}^{C \times C_0}$ and $\mathbf{W}^3 \in \mathbb{R}^{C_0 \times C_0}$, respectively. The output feature maps of the first 1×1 convolution layer, the CFAR filter, and the second 1×1 convolution layer are denoted by $\mathbf{F}^1 \in \mathbb{R}^{M \times N \times C_0}$, $\mathbf{F}^2 \in \mathbb{R}^{M \times N \times C_0}$, $\mathbf{F}^3 \in \mathbb{R}^{M \times N \times C_0}$, respectively. The processing step of S-CFAR block is denoted as

$$\begin{cases} \mathbf{F}^1 = f_{1 \times 1}(\mathbf{X}, C_0) \\ \mathbf{F}^2 = f_{k \otimes g}^{\text{CFAR}}(\mathbf{F}^1) \\ \mathbf{F}^3 = f_{1 \times 1}(\mathbf{F}^2, C_0) \end{cases}, \quad (7)$$

where $f_{1 \times 1}(\cdot, C_0)$ represents 1×1 convolution and batch normalization with C_0 channels. According to [24,25], the p -th channel of $\mathbf{F}^1$ is

$$F_{::,p}^1 = \max \left[\left(\sum_{k=1}^C \mathbf{X}_{::,k} w_{k,p}^1 - \mu_p \right) \frac{\gamma_p}{\sigma_p} + \beta_p, 0 \right], \quad (8)$$

where $0 \leq p \leq C_0 - 1$, μ_p , and σ_p are the values of channel-wise mean and standard deviation of batch normalization, and γ_p and β_p are the learned scaling factor and bias term, respectively.

The CFAR filter is applied to $\mathbf{F}^1$ in channel-by-channel manner. According to Equation (6), the output feature map is

$$F_{::,p}^2 = F_{::,p}^1 - \alpha \bar{F}_{::,p}^1, \quad (9)$$

where $\bar{F}_{::,p}^1$ is the mean value of $F_{::,p}^1$ estimated by CA.

Similar to (8), the d -th channel of the final output feature map is

$$F_{::,d}^3 = \max \left[\left(\sum_{p=1}^{C_0} F_{::,p}^2 w_{p,d}^3 - \mu_d \right) \frac{\gamma_d}{\sigma_d} + \beta_d, 0 \right], \quad (10)$$

From Equations (7)–(10), one can see that different channels of the input are first weighted and interacted. The CFAR filter is applied to obtain the divergence features. At last, the divergence features of different channels are weighted and combined. The nonlinearity of the S-CFAR block is achieved by the ReLU activation function $\max(\cdot, 0)$. Compared to the single-feature CFAR filter, the S-CFAR block can extract multiple nonlinear features.

(2) Multiple-branch CFAR Block

As shown in Figure 4b,c, the IN-CFAR-I block and the IN-CFAR-II block are designed based on the idea of Inception module [26]. Each block consists of three branches with $C_o/2$, $C_o/4$, $C_o/4$ channels. In the IN-CFAR-I block, the first branch is the 1×1 convolution, and the second and third branches are S-CFAR blocks with different sliding windows and guard areas. The output features of these three branches are concatenated and fed into the next layer. In the IN-CFAR-II block, the CFAR filter is replaced by CA operation. The output features of the 1×1 convolution and CA operations are concatenated. Since 1×1 convolution in the next layer weights and combines the features, it can be viewed that the divergence features are implicitly modeled.

The output feature map of the IN-CFAR-I block is given by

$$F = \text{Concat} \left[f_{1 \times 1}(X, C_o/2), f_{k_1 \times g_1}^{S-CFAR}(X, C_o/4), f_{k_2 \times g_2}^{S-CFAR}(X, C_o/4) \right], \quad (11)$$

where $f^{S-CFAR}(\cdot)$ denotes the S-CFAR block which is implemented via Equations (7)–(10).

While the output feature map of the IN-CFAR-II block is given by

$$F = \text{Concat} \left[f_{1 \times 1}(X, C_o/2), f_{k_1 \times g_1}^{S-CA}(X, C_o/4), f_{k_2 \times g_2}^{S-CA}(X, C_o/4) \right], \quad (12)$$

where $f^{S-CA}(\cdot)$ is similar to S-CFAR block but with the CFAR filter replaced by CA operation.

3.2. Deep CFAR Networks

Similar to classical deep networks such as A-ConvNets [7] and ResNet [24], the proposed deep CFAR networks are composed of a feature extraction part and a classifier. The feature extraction part contains four stages, as shown in Figure 5. In each stage, a CFAR block and a max pooling layer with stride 2 are used. The average pooling layer and fully connection layer are used as a classifier, where the number of output channels is 2. The class probabilities of target and clutter are given by the Softmax function.

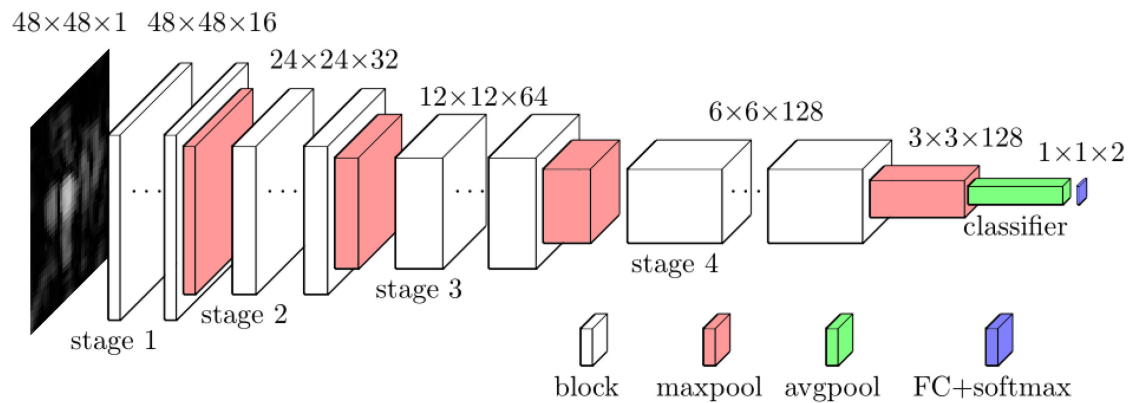


Figure 5. Proposed CFAR block-based network architectures. A 48×48 single-channel SAR image is taken as an example.

Based on the CFAR blocks, three deep CFAR networks named A-CFARNet, B-CFARNet, and C-CFARNet are proposed. The network specifications are shown in Table 1. In the first stage, these three networks use a large sliding window and a guard area. In the last three stages, the sizes of sliding window and guard area are decreased. To keep the simplicity of the network, we did not exhaust all the possible sizes. Instead, we mainly use two scales, i.e., one with a large receptive field and one with a small receptive field. In A-CFARNet, 17×9 and 5×3 are used. In B-CFARNet and C-CFARNet, 11×7 and 7×5 are added.

Table 1. The network specifications of CFARNets.

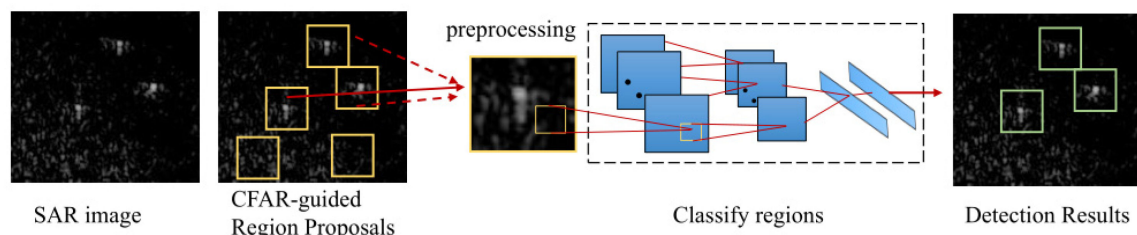
Stages	A-CFARNet	B-CFARNet	C-CFARNet
stage 1	S-CFAR 17□9	IN-CFAR-I 17□9 11□7	IN-CFAR-II 17□9 11□7
stage 2	S-CFAR 5□3	IN-CFAR-I 7□5 15□3	IN-CFAR-II 7□5 15□3
stage 3	S-CFAR 5□3	IN-CFAR-I 7□5 15□3	IN-CFAR-II 7□5 15□3
stage 4	S-CFAR 5□3	IN-CFAR-I 7□5 15□3	IN-CFAR-II 7□5 15□3

4. Two-Stage Target Detection Based on Deep CFAR Networks

Target detection mainly includes two stages. First, possible target regions are generated by methods such as sliding window, traditional detection methods such as CFAR, and networks. Second, statistics or classifiers are applied to determine whether the candidates are targets or not. In this work, a two-stage target detection method is proposed by combining CA-CFAR and the proposed CFARNets.

4.1. Two-Stage Detection Framework

Figure 6 shows the proposed two-stage target detection framework. For an input SAR image, the CA-CFAR is first applied. Region proposals are generated based on the CFAR results. Next, region proposals are preprocessed and fed into CFARNets to determine whether they are target or clutter. Finally, results fusion is carried out to improve the detection results.

**Figure 6.** Two-stage target detection framework.

4.2. CFAR-Guided Region Proposals

Here, the two-parameter CA-CFAR is used [27]. It is derived based on the assumption that the clutter is Gaussian distributed. The detection rule is given by

$$\frac{z(i,j) - \bar{Z}(i,j)}{\bar{\sigma}(i,j)} \underset{H_0}{\overset{H_1}{\gtrless}} \alpha, \quad (13)$$

where $\bar{Z}(i,j)$ and $\bar{\sigma}(i,j)$ are the estimated mean value and standard deviation of the clutter patch in the sliding window, respectively. α is the detection threshold which is determined by the probability of false alarms. The probability of false alarms for a given α is

$$P_{FA}(\alpha) = \int_{\alpha}^{+\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{t^2}{2}} dt.$$

After CFAR detection, clustering of the detected pixels is carried out based on their distances. Each cluster is represented by the pixel with maximum amplitude, which is denoted by (x_i^c, y_i^c) , $i \in [1, N_0]$ is the cluster index, N_0 is the number of clusters (targets). Region proposals with a size of 48×48 based on the cluster centers were generated.

4.3. Multi-Crop Classification

Region proposals were resized to 55×55 and their center crops with a size of 48×48 were fed into the networks. This is referred to as standard mode. In addition

to the standard mode, multi-crop classification, which provides flexible controlling of detection quality, was employed as well. As shown in Figure 7, for each region proposal, the center region and two random regions of the resized image are cropped out. These three crops are fed into CFARNet, and the classification results are fused.

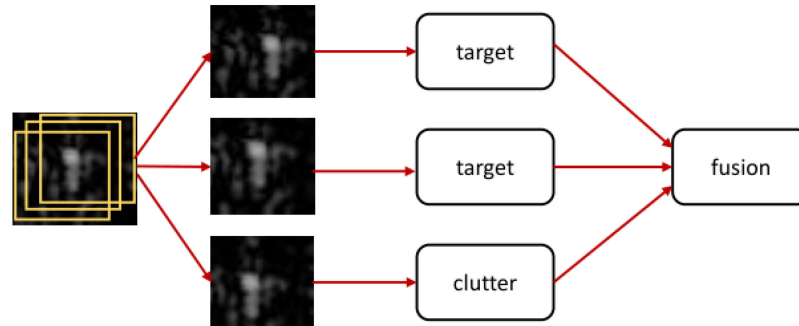


Figure 7. Multi-crop classification.

Denote target probabilities of the three crops by y_1 , y_2 , and y_3 . Two fusion strategies are used.

- (1) Eager mode. A region proposal is judged as a target if one of the three crops is a target, i.e., the maximum target probability is larger than 0.5.

$$\max\{y_1, y_2, y_3\} \underset{H_0}{\overset{H_1}{\geq}} 0.5, \quad (14)$$

- (2) Steady mode. A region proposal is judged as a target if the mean target probability of the three crops is larger than 0.5.

$$\text{mean}\{y_1, y_2, y_3\} \underset{H_0}{\overset{H_1}{\geq}} 0.5, \quad (15)$$

5. Experimental Results

5.1. Dataset

The detection performance was evaluated by measured UWB SAR images. Figure 8 shows the four SAR images collected by the airship-mounted UWB SAR (AMUSAR) system [4]. The range resolution and azimuth resolution of the SAR images is $0.15 \text{ m} \times 0.15 \text{ m}$. The illuminated areas mainly contain mountains, farmland, bare soil, and roads. The first two SAR images have almost the same illuminated area but with different landmine settings. These images are collected by different flights. The imaging conditions are not identical, as is the case for the imaging qualities of the landmines.

These four SAR images are divided into image patches with a size of 512×512 . There are 61 image patches with landmines. These image patches are used to construct the detection dataset, which is denoted as AMUDet. The training set contains 55 image patches, while the test set contains six image patches. Figure 9 shows examples of image patches extracted from these four SAR images. The bright point-shaped targets are landmines. It can be observed that these landmines vary as the image conditions change. The landmines in the collected SAR images are annotated manually. A summary of AMUDet is provided in Table 2.

5.2. Experimental Setup

The performance of the proposed two-stage detection method was evaluated based on AMUDet. The two-parameter CA-CFAR detector was used for region proposal generation.

The sizes of the sliding window and guard area are 63×63 and 55×55 , respectively. The probability of false alarms is set as 10^{-3} . The deep networks are trained based on image chips of AMUDet. Landmine chips are cropped from images based on the annotated rectangles. For each image patch, 20 randomly extracted regions with a size of 48×48 were taken as clutter chips. The processing flow of the experiment is given in Figure 10.

Table 2. Details of AMUDet.

Dataset	Train	Test
NO. of images	55	6
NO. of landmines	297	58

For comparison, TinyResNet-18, A-ConvNets48, and Conv1x1Net were used in the proposed two-stage detection method as well. The TinyResNet-18 and A-ConvNets48 are customized to fit the input 48×48 image based on ResNet-18 [24] and A-ConvNets [7]. The Conv1x1Net is a stacking of 1×1 convolution layers. Details of these three networks are provided in Appendix A.

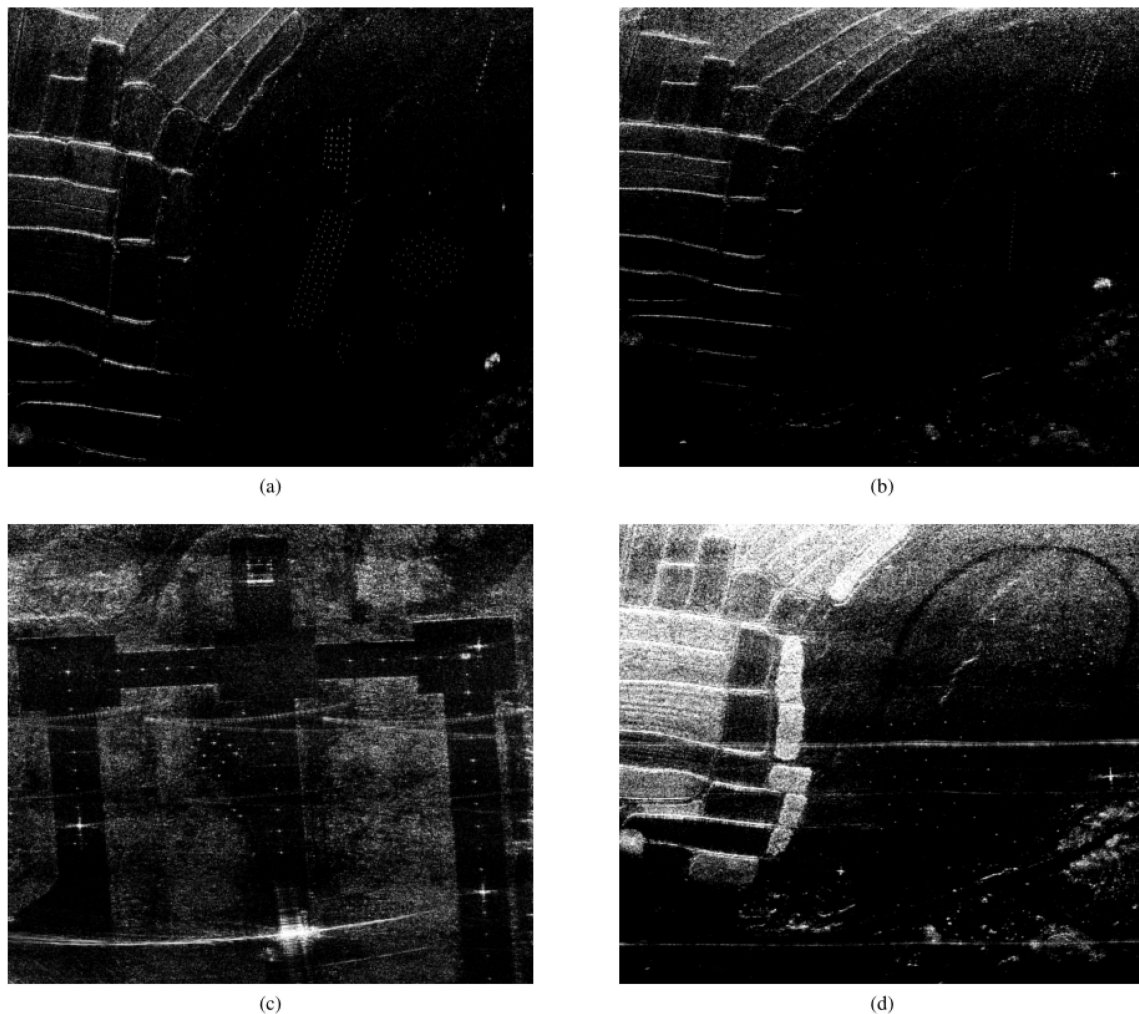


Figure 8. Collected SAR images with landmines. The sizes of images are: (a) 3751×5002 ; (b) 5001×5001 ; (c) 3166×2643 ; (d) 5001×5001 .

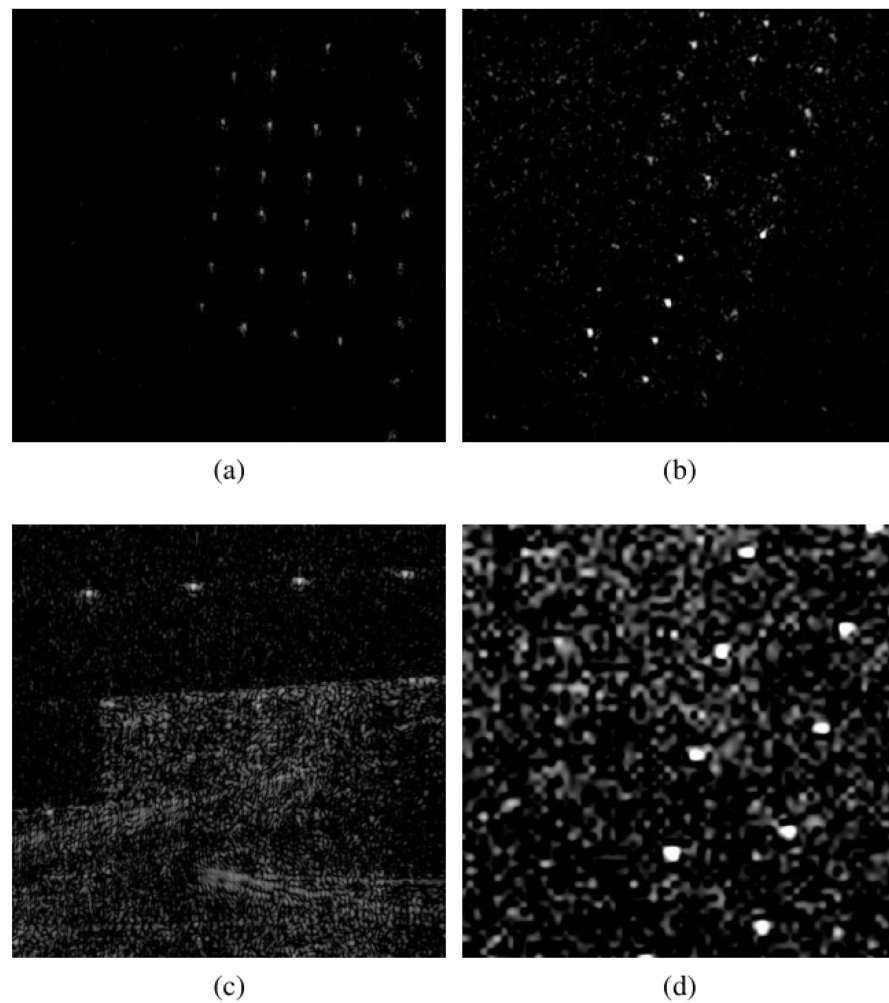


Figure 9. Examples of image patches: (a–d) Local part of minefield.

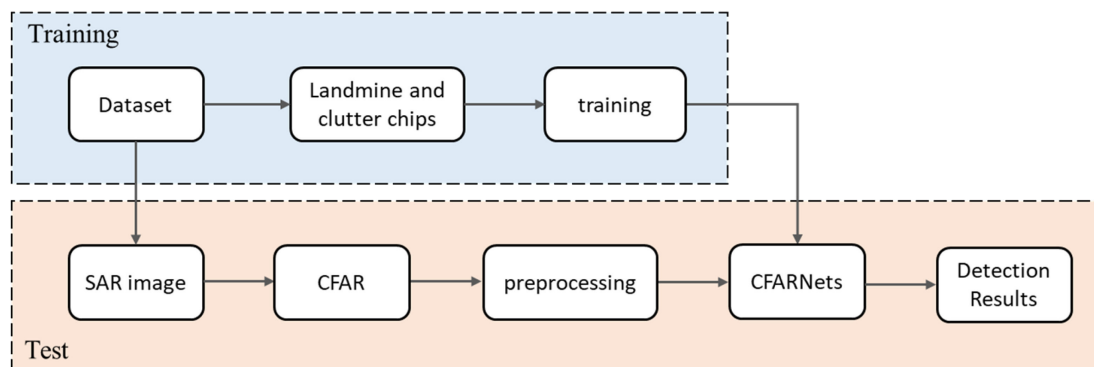


Figure 10. Processing flow of the experiment.

These models are trained by stochastic gradient descent (SGD) with a batch size of 128 examples, momentum of 0.9, and weight decay of 0.0005. The initial learning rate is 0.01 and is divided by 10 at the 30 and 45 epochs. Overall, 60 epochs were used for training. During training, the following preprocessing pipelines were used:

- (1) Randomly crop a rectangular region whose aspect ratio is randomly sampled in $[3/4, 4/3]$ and area randomly sampled in $[80\%, 100\%]$, then resize the cropped region into a 48-by-48 square image.
- (2) Flip horizontally with 0.5 probability.

- (3) Scale brightness with coefficients uniformly drawn from [0.6, 1.4].
- (4) Normalize the gray image by subtracting 0.5.

During validation, the shorter edge of the image is resized to 55 with a fixed aspect ratio. Following that, the resized image is cropped and normalized.

5.3. Detection Results

- (1) The proposed two-stage detection method

Detection performances with different modes are shown in Tables 3–5. Since the proposed two-stage detection methods combine CFAR and deep networks, these methods are denoted as CFAR-{networks}. The CFAR detector has a high recall of 98.28%, but its precision and F1 score are low. Equipped with feature learning ability, CFAR-A-ConvNets48, CFAR-TinyResNet-18, CFAR-A-CFARNet, CFAR-B-CFARNet, and CFAR-C-CFARNet have a much larger F1 score due to the increase of precision. Among the three detection modes, the overall F1 score of the steady mode is the highest, while the eager mode has the highest precision, and the standard mode has a median F1 score (recall) and precision. Based on these results, the detection mode can be chosen according to the preference of high recall or precision.

Table 3. Landmine detection performance with standard mode.

Method	Recall (%)	Precision (%)	F1 (%)
Two-parameter CA-CFAR	98.28	53.27	69.09
CFAR-Conv1x1Net	51.72	62.50	56.60
CFAR-A-ConvNets48	98.28	71.25	82.61
CFAR-TinyResNet-18	96.55	72.73	82.96
CFAR-A-CFARNet	94.83	75.34	83.97
CFAR-B-CFARNet	93.10	75.00	83.08
CFAR-C-CFARNet	91.38	81.54	86.18

Table 4. Landmine detection performance with eager mode.

Method	Recall (%)	Precision (%)	F1 (%)
CFAR-Conv1x1Net	63.79	56.06	59.68
CFAR-A-ConvNets48	98.28	65.52	78.62
CFAR-TinyResNet-18	98.28	70.37	82.01
CFAR-A-CFARNet	96.55	72.73	82.96
CFAR-B-CFARNet	96.55	70.89	81.75
CFAR-C-CFARNet	91.38	76.81	83.46

Table 5. Landmine detection performance with steady mode.

Method	Recall (%)	Precision (%)	F1 (%)
CFAR-Conv1x1Net	44.83	60.47	51.49
CFAR-A-ConvNets48	98.28	73.08	83.82
CFAR-TinyResNet-18	96.55	76.71	85.50
CFAR-A-CFARNet	93.10	83.08	87.80
CFAR-B-CFARNet	96.55	75.68	84.85
CFAR-C-CFARNet	89.66	82.54	85.95

When the standard mode is used, CFAR-C-CFARNet has the highest F1 score, 86.18%, followed by CFAR-A-CFARNet and CFAR-B-CFARNet. When the eager mode is used, CFAR-C-CFARNet has the highest F1 score, 83.46%, followed by CFAR-A-CFARNet, CFAR-TinyResNet-18, and CFAR-B-CFARNet. When the steady mode is used, CFAR-A-CFARNet has the highest F1 score, 87.80%, followed by CFAR-C-CFARNet, CFAR-TinyResNet-18, and CFAR-B-CFARNet. We can also see that the average F1 score of proposed CFAR-CFARNets

is 1.8%, 4.1%, and 2.38%, higher than the F1 score of CFAR-A-ConvNets48 in standard mode, eager mode, and steady mode, respectively.

Figure 11 shows the landmine detection results of CFAR-A-CFARNet. There are four missed targets and 11 false alarms. The false alarms are mainly concentrated on the fifth image, which has seven false alarms.

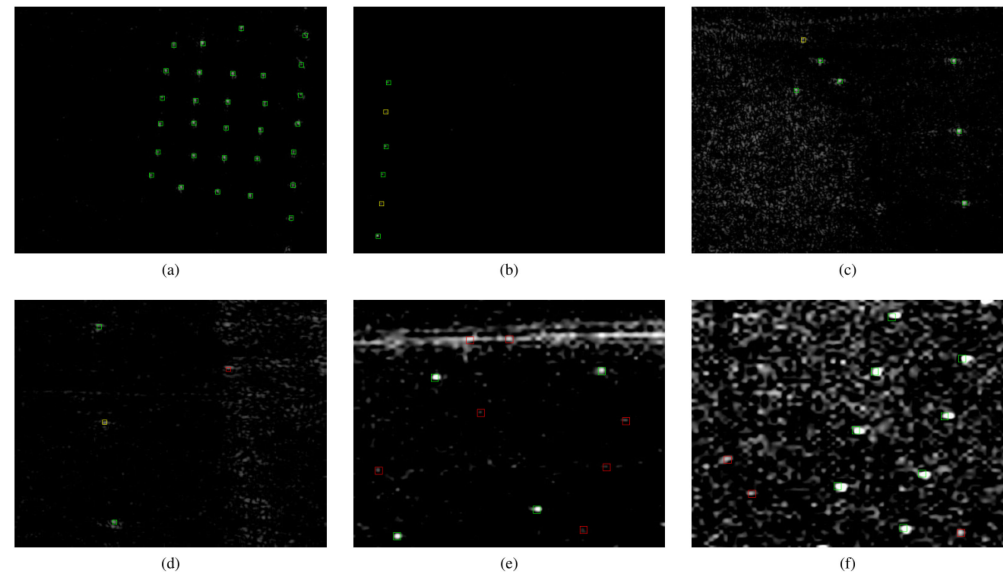


Figure 11. Landmine detection results of CFAR-A-CFARNet. (a–f) Local part of minefield. The rectangles indicate the detection results: Green rectangle, detected target; Red rectangle, false alarm; Yellow rectangle, missed target.

(2) Comparison with other detectors

Faster R-CNN [28] and real-time state-of-art detectors, including YOLOv3 [29], YOLOX [30], and RTMDet [31], were used for comparison as well. The input sizes of all these detectors are 512×512 . Their results are shown in Table 6. The F1 scores of CFAR-C-CFARNet in standard mode and steady mode, and CFAR-A-CFARNet in steady mode, are comparable to that of YOLOv3. All the proposed detectors outperformed YOLOX-S, YOLOX-Tiny, RTMDet-S, and RTMDet-Tiny in terms of F1 score. Although Faster R-CNN had a higher F1 score compared to the proposed method, its inference speed was the lowest.

Table 6. Landmine detection performance of other detectors.

Method	Recall (%)	Precision (%)	F1 (%)	Latency
Faster R-CNN	96.55	81.16	88.19	165.6
YOLOv3	98.28	77.03	86.36	24.3
YOLOX-S	94.83	68.75	79.71	23.5
YOLOX-Tiny	68.97	78.43	73.39	22.2
RTMDet-S	93.10	75.00	83.08	25.2
RTMDet-Tiny	91.38	69.74	79.10	23.4

Figure 12 plots the speed and accuracy of the real-time detectors. Only the real-time detectors are included. The speed is measured with FP32-precision and batch = 1 on a single Tesla V100. Among these detectors, the proposed CFAR-guided two-stage detection methods are much faster than YOLOs and RTMDets. Remarkably, CFAR-A-CFARNet has a comparable F1 score with YOLOv3 but only one-half latency.

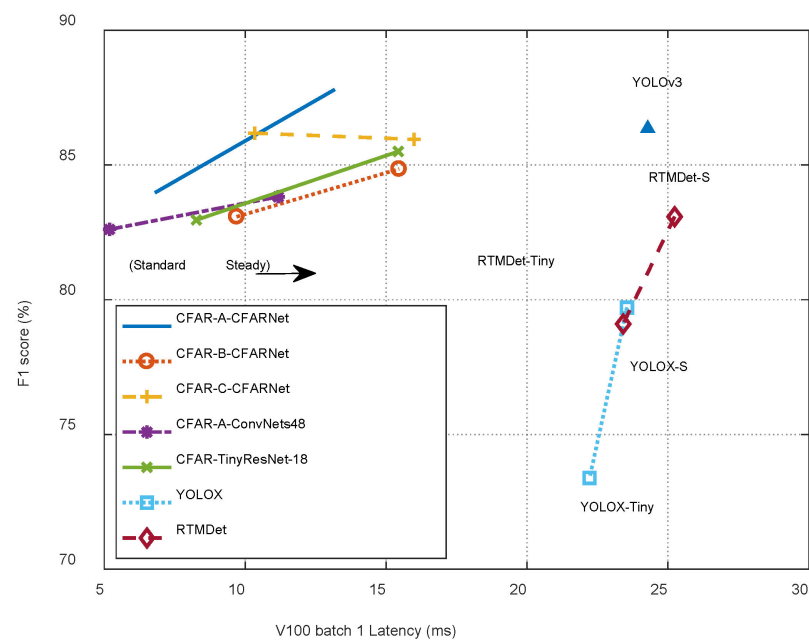


Figure 12. Speed–accuracy curve of the detectors.

The number of parameters and FLOPS for these detectors are shown in Table 7, where N_T and N_C are the target number of CFAR detectors and the crop number of image chips, respectively. Herein, the average detected target number per image of the test set is 17.8. Taking $N_T = 17.8$ and $N_C = 3$ for example, the computational cost of the proposed CFAR-X-CFARNets detectors is less than 1 GFLOPS.

Table 7. Number of parameters and FLOPS.

Model	#Params (M)	FLOPs (G)
Conv1x1Net	0.162	$0.003 \times N_T \times N_C$
A-ConvNets48	1.371	$0.023 \times N_T \times N_C$
TinyResNet-18	11.681	$0.315 \times N_T \times N_C$
A-CFARNet	0.162	$0.004 \times N_T \times N_C$
B-CFARNet	0.143	$0.002 \times N_T \times N_C$
C-CFARNet	0.143	$0.002 \times N_T \times N_C$
Faster R-CNN	32.963	758
YOLOv3	61.524	48.534
YOLOX-S	8.938	8.524
YOLOX-Tiny	5.033	4.845
RTMDet-S	8.856	9.440
RTMDet-Tiny	4.873	5.136

(3) Influence of receptive field

The influence of receptive field, i.e., the sizes of sliding window and guard area, was evaluated. The receptive field of A-CFARNet is set as 5×3 , 7×5 , 9×5 , 11×7 , 13×7 , and 17×9 . For each receptive field, the model is trained from scratch 5 times, and the detection metrics are averaged. Figure 13 plots the detection metrics of CFAR-A-CFARNet. The changes in detection metrics with different receptive fields are smaller than 3.8%. The maximum F1 score is achieved when the receptive field is 13×7 , which is 86.27%. When the receptive field is 7×5 , 11×7 or 17×9 , the F1 score is about 84.6%.

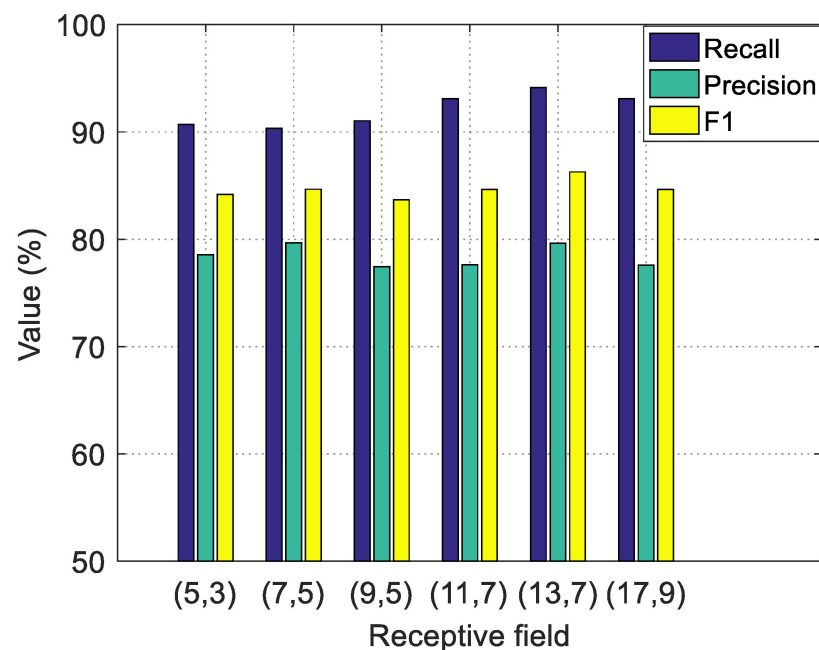


Figure 13. Detection metrics with different receptive fields.

(4) Influence of CFAR parameter

As CFAR is used to generate target proposals, its parameter setting may affect the detection performance of the proposed method. To evaluate the influence, the probability of false alarms of CFAR was set from 10^{-6} to 10^{-2} . The detection metrics of CFAR and the proposed method are plotted in Figure 14. It can be seen that the recall of CFAR increases when P_{FA} increases, and the precision has a converse trend. As the F1 score is the weighted sum of recall and precision, its value first increases when P_{FA} is small and then decreases when is P_{FA} large. The detection metrics of CFAR-A-CFARNet, CFAR-B-CFARNet, and CFAR-C-CFARNet have similar trends with CFAR. The CFAR-A-CFARNet has a maximum value of F1 score 85.45% at $P_{FA} = 10^{-4}$, the CFAR-B-CFARNet has a maximum value of F1 score 84.03% at $P_{FA} = 10^{-3.75}$, and the CFAR-A-CFARNet has a maximum value of F1 score 86.67% at $P_{FA} = 10^{-2.75}$. Note that the F1 scores of CFAR-A-CFARNets are relatively stable when $10^{-4} \leq P_{FA} \leq 10^{-3}$ and decrease after $P_{FA} = 10^{-3}$. As the recall at $P_{FA} = 10^{-3}$ is larger than that at $P_{FA} = 10^{-4}$, it is suggested that we use the setting of $P_{FA} = 10^{-3}$ for CFAR to obtain a high F1 score and high recall.

5.4. Feature Analysis

To further obtain insight into the mechanism of the proposed CFARNets, a landmine image and a clutter image are fed into the proposed CFARNets, and their extracted features at each stage are analyzed. These two example images are plotted in Figure 15.

Figures 16 and 17 show the extracted features of A-CFARNet for these two example images. It can be found that there are two types of the first two stage features: the first type of features has similar shapes to the input images while the second type of features has shapes that are complementary to the input images. It can also be seen that the features have a decreasing resolution from the second stage to the fourth stage and the semantic information is increasing. According to these results and the mathematical property of CFARNets, we can infer that the CFARNets mainly extract images' divergent features.

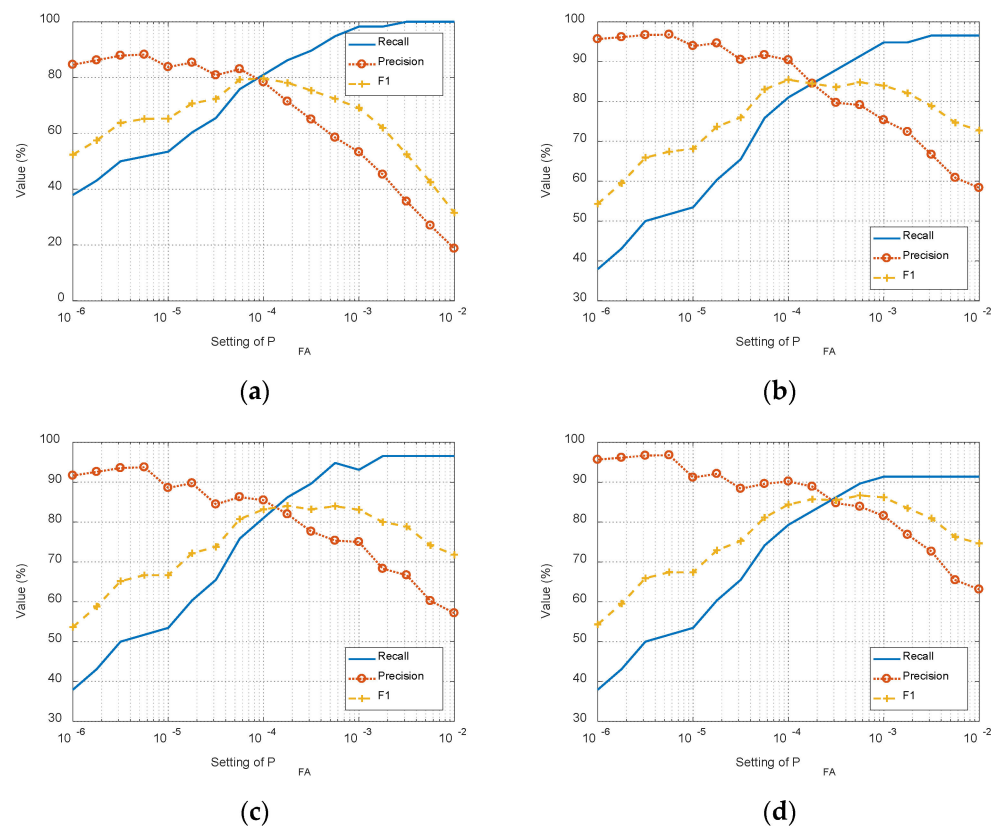


Figure 14. Detection metrics with different probabilities of false alarms. (a) CFAR; (b) CFAR-A-CFARNet; (c) CFAR-B-CFARNet; (d) CFAR-C-CFARNet.

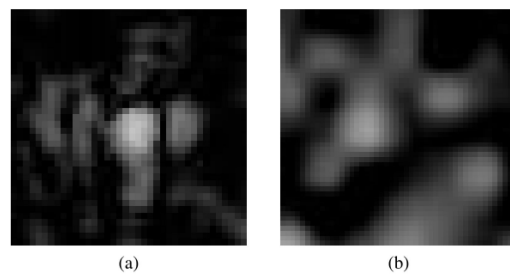


Figure 15. Examples of image patches. (a) landmine; (b) clutter.

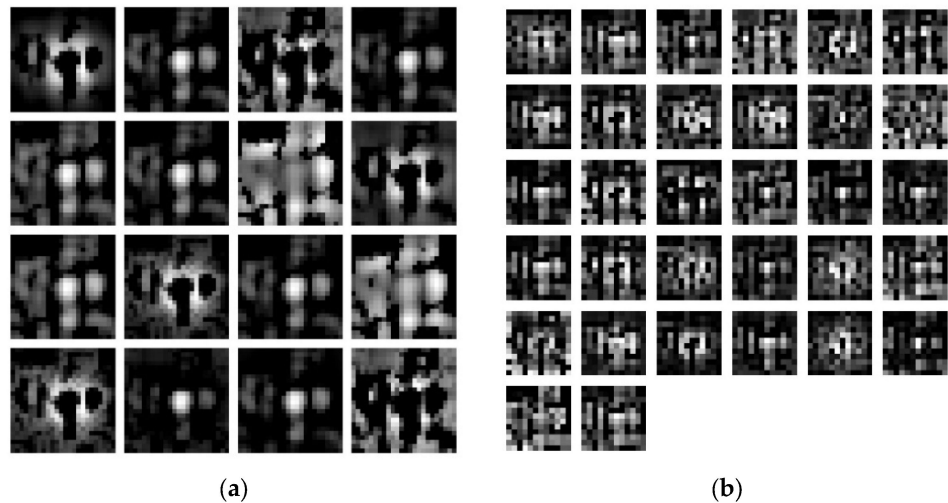


Figure 16. Cont.

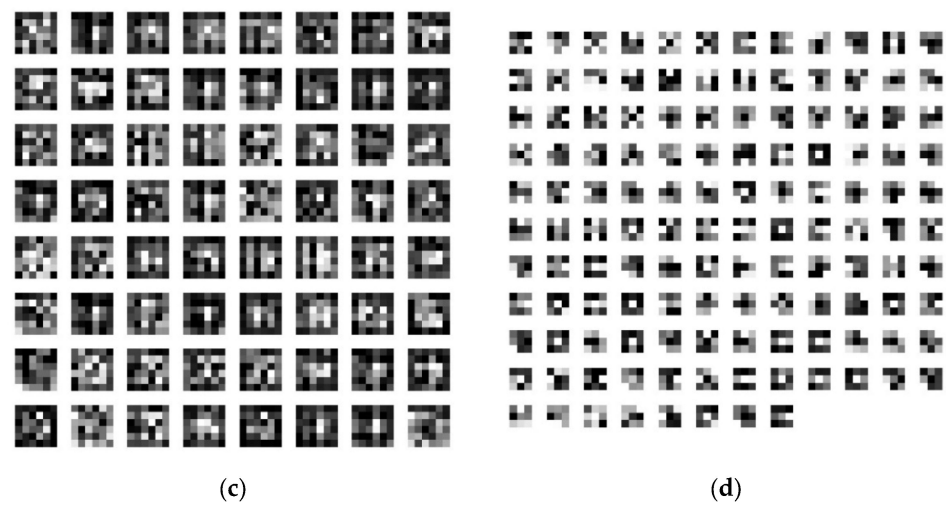


Figure 16. Feature maps of the landmine image. (a) stage 1; (b) stage 2; (c) stage 3; (d) stage 4.

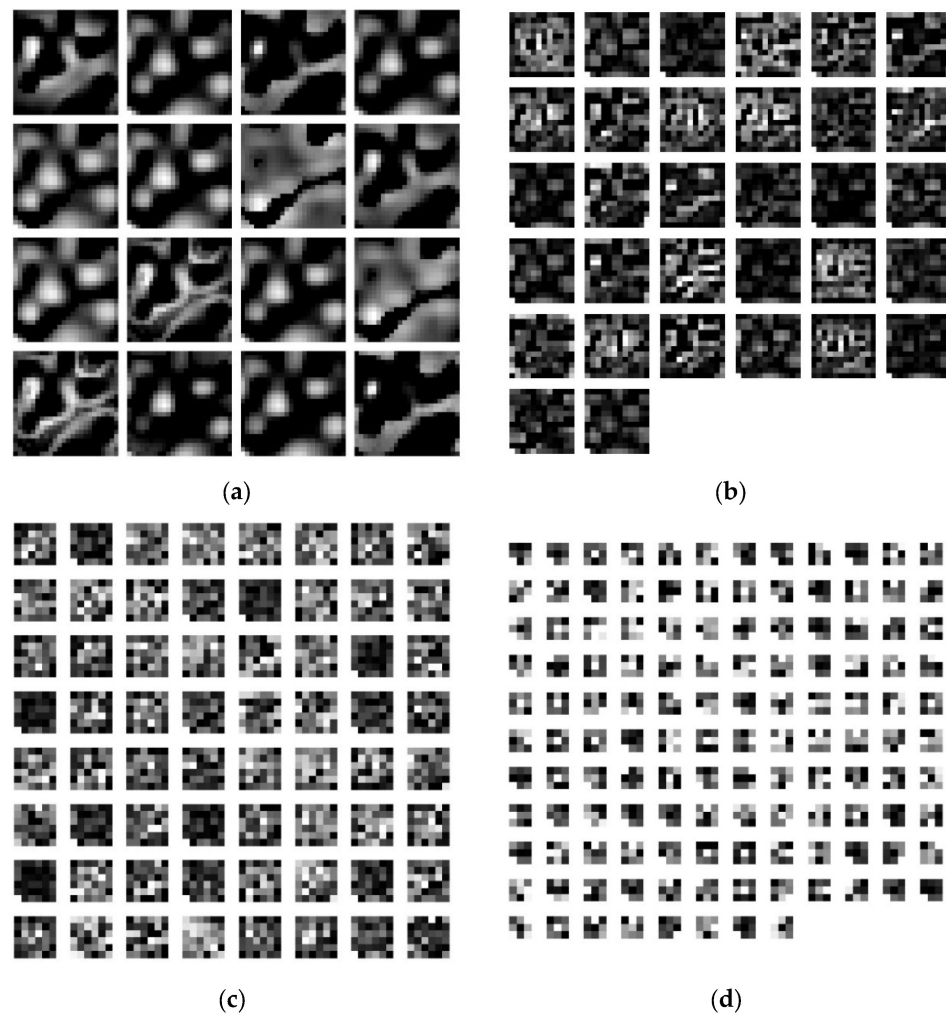


Figure 17. Feature maps of the clutter image. (a) stage 1; (b) stage 2; (c) stage 3; (d) stage 4.

6. Discussion

The high-resolution image obtained by UWB SAR includes rich features of landmines, which is helpful for landmine discrimination and detection. This work not only proposes lightweight deep networks based on CFAR filters but also studies the detection performances of other deep detectors. The experimental results suggest that:

- (1) Just using the nonlinearity of 1×1 convolution layer is insufficient to build a good network model. Other filters are essential.
- (2) By combining the CFAR filters and the 1×1 convolution layer, the image's multi-dimensional divergence features are extracted layer-by-layer, and we can obtain a high-performance network for SAR landmine detection.
- (3) Compared to other real-time state-of-the-art detectors, the proposed CFARNets have comparable performance in terms of F1 score, with a significant reduction in the number of parameters and flops.

7. Conclusions

In this work, we propose a new class of filter, block, and networks that can serve as an alternative to that in convolutional neural networks (CNNs) for landmine detection in SAR images. It was first shown that the CA-CFAR detector can be implemented by using tensor operations, which makes it possible to be used as an interpretable network module. Three CFAR blocks and three CFARNets were proposed by integrating classical network micro-architectures and nonlinear filters. Furthermore, a two-stage landmine detection method based on CFARNets was proposed. The features extracted by CFARNets are interpretable, as CFAR filters have definite physical significance. The proposed CFARNets can efficiently utilize the SAR characteristic, and their detection performances are comparable to YOLO detectors with a significant reduction in the number of parameters and flops. Although the proposed CFARNets have a high performance for landmine detection, their performance on other targets remains to be tested. The proposed two-stage detection method depends on the CFAR detector whose parameters are manually designed. Combining the CFAR detector and CFARNets as a whole network would be the key point of future research.

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

Funding: This research is co-supported by the National Natural Science Foundation of China (Grant No. 61971430), and the Natural Science Foundation of Hunan Province, China (Grant No.2021JJ40358).

Data Availability Statement: Not applicable.

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

Appendix A

Appendix A.1. TinyResNet-18

The stride 2, 7×7 kernel of the first convolution layer in ResNet-18 is modified as stride 1, 3×3 kernel. Table A1 shows the network architecture of TinyResNet-18.

Table A1. TinyResNet-18 architecture for landmine.

Layer Name	Output Size	Parameter Setting
conv1	48×48	3×3 , 64, stride 1
conv2_x	24×24	3×3 max pool, stride 2
conv3_x	12×12	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$
conv4_x	6×6	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$
conv5_x	3×3	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2$
conv5_x	3×3	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$
classifier	1×1	average pool, 2-d fc, softmax

Appendix A.2. A-ConvNets48

The A-ConvNets48 contains five convolution layers. The first three convolution layers use zero padding to keep their output sizes, while the 5×5 kernel in the fourth convolution layer is modified as a 4×4 kernel. Table A2 shows the network architecture of TinyResNet-18.

Table A2. A-ConvNets48 architecture for landmine.

Layer Name	Output Size	Parameter Setting
conv1	48×48	$5 \times 5, 16$
maxpool1	24×24	2×2
conv2	24×24	$5 \times 5, 32$
maxpool2	12×12	2×2
conv3	12×12	$6 \times 6, 64$
maxpool3	6×6	2×2
conv4	3×3	$4 \times 4, 128$
conv5	1×1	$3 \times 3, 2$

Appendix A.3. Conv1x1Net

Similar to the proposed CFARNets, the Conv1x1Net has 4 stages as well but with the CFAR blocks replaced by 1×1 convolution layers. There are 1, 2, 2, and 2 convolution layers in each stage. The output channel at the end of each stage is the same as the proposed CFARNets.

References

1. Song, X.; Xiang, D.; Zhou, K.; Su, Y. Fast prescreening for gpr antipersonnel mine detection via go decomposition. *IEEE Geosci. Remote. Sens. Lett.* **2019**, *16*, 15–19. [\[CrossRef\]](#)
2. Temlioglu, E.; Erer, I. A novel convolutional autoencoder-based clutter removal method for buried threat detection in ground-penetrating radar. *IEEE Trans. Geosci. Remote. Sens.* **2022**, *60*, 1–13. [\[CrossRef\]](#)
3. Jin, T.; Zhou, Z. Feature extraction and discriminator design for landmine detection on double-hump signature in ul-trawideband sar. *IEEE Trans. Geosci. Remote Sens.* **2008**, *46*, 3783–3791. [\[CrossRef\]](#)
4. Lou, J.; Jin, T.; Liang, F.; Zhou, Z. A novel prescreening method for land-mine detection in UWB SAR based on feature point matching. *IEEE Trans. Geosci. Remote Sens.* **2013**, *51*, 3706–3714. [\[CrossRef\]](#)
5. Zhang, X.; Yang, P.; Zhou, M. Multireceiver SAS imagery with generalized PCA. *IEEE Geosci. Remote Sens. Lett.* **2023**, *20*, 1–5. [\[CrossRef\]](#)
6. Hu, X.; Xie, H.; Zhang, L.; Hu, J.; He, J.; Yi, S.; Jiang, H.; Xie, K. Fast Factorized Backprojection Algorithm in Orthogonal Elliptical Coordinate System for Ocean Scenes Imaging Using Geosynchronous Spaceborne–Airborne VHF UWB Bistatic SAR. *Remote Sens.* **2023**, *15*, 2215. [\[CrossRef\]](#)
7. Chen, S.; Wang, H.; Xu, F.; Jin, Y.-Q. Target classification using the deep convolutional networks for SAR images. *IEEE Trans. Geosci. Remote Sens.* **2016**, *54*, 4806–4817. [\[CrossRef\]](#)
8. Wagner, S.A. SAR ATR by a combination of convolutional neural network and support vector machines. *IEEE Trans. Aerosp. Electron. Syst.* **2016**, *52*, 2861–2872. [\[CrossRef\]](#)
9. Yang, H.; Zhang, T.; He, Y.; Dan, Y.; Yin, J.; Ma, B.; Yang, J. GPU-oriented designs of constant false alarm rate detectors for fast target detection in radar images. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 1–14. [\[CrossRef\]](#)
10. Yu, W.; Wang, Y.; Liu, H.; He, J. Superpixel-based CFAR target detection for high-resolution SAR images. *IEEE Geosci. Remote Sens. Lett.* **2016**, *13*, 730–734. [\[CrossRef\]](#)
11. Pappas, O.; Achim, A.; Bull, D. Superpixel-level CFAR detectors for ship detection in SAR imagery. *IEEE Geosci. Remote Sens. Lett.* **2018**, *15*, 1397–1401. [\[CrossRef\]](#)
12. Li, M.-D.; Cui, X.-C.; Chen, S.-W. Adaptive superpixel-level CFAR detector for SAR inshore dense ship detection. *IEEE Geosci. Remote Sens. Lett.* **2021**, *19*, 1–5. [\[CrossRef\]](#)
13. Chabbi, S.; Farah, F.; Guidoum, N. CFAR-CNN detector of ships from SAR image using generalized gamma distribution and real dataset. In Proceedings of the 2022 7th International Conference on Image and Signal Processing and Their Applications (ISPA), Mostaganem, Algeria, 8–9 May 2022; pp. 1–6.
14. Tang, T.; Wang, Y.; Liu, H.; Zou, S. CFAR-guided dual-stream single-shot multibox detector for vehicle detection in SAR images. *IEEE Geosci. Remote Sens. Lett.* **2022**, *19*, 1–5. [\[CrossRef\]](#)
15. Torriero, P.A.; Morton, K.D.; Sakaguchi, R.; Collins, L.M. Histograms of oriented gradients for landmine detection in ground-penetrating radar data. *IEEE Trans. Geosci. Remote Sens.* **2014**, *52*, 1539–1550. [\[CrossRef\]](#)

16. Kwak, Y.; Song, W.-J.; Kim, S.-E. Speckle-noise-invariant convolutional neural network for SAR target recognition. *IEEE Geosci. Remote Sens. Lett.* **2019**, *16*, 549–553. [[CrossRef](#)]
17. Wang, N.; Wang, Y.; Liu, H.; Zuo, Q.; He, J. Feature-fused SAR target discrimination using multiple convolutional neural Networks. *IEEE Geosci. Remote Sens. Lett.* **2017**, *14*, 1695–1699. [[CrossRef](#)]
18. Cui, J.; Jia, H.; Wang, H.; Xu, F. A fast threshold neural network for ship detection in large-scene SAR images. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2022**, *15*, 6016–6032. [[CrossRef](#)]
19. Wang, Z.; Du, L.; Mao, J.; Liu, B.; Yang, D. SAR target detection based on SSD with data augmentation and transfer learning. *IEEE Geosci. Remote Sens. Lett.* **2019**, *16*, 150–154. [[CrossRef](#)]
20. 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](#)]
21. Lang, P.; Fu, X.; Feng, C.; Dong, J.; Qin, R.; Martorella, M. Lw-cmdanet: A novel attention network for SAR automatic target recognition. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2022**, *15*, 6615–6630. [[CrossRef](#)]
22. Tai, Y.; Tan, Y.; Xiong, S.; Sun, Z.; Tian, J. Few-Shot transfer learning for SAR image classification without extra SAR samples. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2022**, *15*, 2240–2253. [[CrossRef](#)]
23. Zhang, W.; Zhu, Y.; Fu, Q. Semi-Supervised deep transfer learning-based on adversarial feature learning for label limited SAR target recognition. *IEEE Access* **2019**, *7*, 152412–152420. [[CrossRef](#)]
24. 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, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 770–778.
25. Ding, X.; Guo, Y.; Ding, G.; Han, J. ACNet: Strengthening the Kernel Skeletons for Powerful CNN via Asymmetric Convolution Blocks. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019. [[CrossRef](#)]
26. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015.
27. Gao, G.; Kuang, G.; Zhang, Q.; Li, D. Fast detecting and locating groups of targets in high-resolution SAR images. *Pattern Recognit.* **2007**, *40*, 1378–1384. [[CrossRef](#)]
28. 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](#)]
29. Redmon, J.; Farhadi, A. Yolo v3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
30. Ge, Z.; Liu, S.; Wang, F.; Li, Z.; Sun, J. Yolo x: Exceeding yolo series in 2021. *arXiv* **2021**, arXiv:2107.08430.
31. Lyu, C.; Zhang, W.; Huang, H.; Zhou, Y.; Wang, Y.; Liu, Y.; Zhang, S.; Chen, K. RTMDet: An Empirical Study of Designing Real-Time Object Detectors. *arXiv* **2022**, arXiv:2212.07784.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.