

Article

F²DN-CCWL: Progressive Sub-Pixel-Level Intelligent Detection for Low Observable Targets in Radar Range-Doppler Spectra

Mingjie Qiu ^{1,2,*}, Jianming Wang ¹ and Guangxin Wu ¹

¹ Nanjing Research Institute of Electronics Technology, No. 8 Guorui Road, Yuhuatai District, Nanjing 210039, China; wangjianming@cetc.com.cn (J.W.); wuguangxin@cetc.com.cn (G.W.)

² China Academy of Electronics and Information Technology, No. 11 Shuangyuan Road, Shijingshan District, Beijing 100041, China

* Correspondence: qiumingjie@cetc.com.cn

Abstract

Aiming at core bottlenecks in weak and small target detection in radar range-Doppler (RD) spectra under low signal-to-noise ratio (SNR)—including severe performance degradation of traditional constant false alarm rate (CFAR) detectors and the inherent trade-off difficulty faced by existing deep learning methods in balancing detection accuracy, localization precision, and real-time performance—this paper proposes a progressive sub-pixel-level intelligent detection algorithm named F²DN-CCWL. The algorithm constructs a three-stage detection pipeline: global candidate screening, local fine discrimination, and weighted localization, and implements a full-stack customized design covering network architecture, soft-label training strategy, and post-processing modules. Simulation and field-measured results demonstrate that at −20 dB SNR, the proposed algorithm achieves a detection probability of 95.3%, a false alarm rate of 3.1%, an average localization error of 0.76 pixels, and a single-frame inference latency of 47.21 ms. This method offers a high-performance engineering solution for radar-based detection of low observable targets.

Keywords: constant false alarm rate (CFAR); deep learning; range-Doppler spectrum; sub-pixel localization; low observable target detection

1. Introduction

Radar target detection is the core functional module of modern pulse-Doppler radar systems, and its performance directly determines the mission effectiveness across a wide range of defense and civilian applications, including low-altitude anti-UAV operations, battlefield early warning, border surveillance, and short-range air defense [1–3]. In recent years, with the widespread deployment of low-altitude, slow-speed, small (LSS) unmanned aerial vehicles (UAVs) and stealth aerial platforms, the radar cross-section (RCS) of detected targets has drastically decreased, resulting in typical received echo SNR values ranging from −20 dB to 0 dB. Consequently, targets manifest as only a few pixels submerged in strong ground clutter and noise, posing a severe performance challenge to conventional detection algorithms.

As the fundamental data representation for radar moving target detection, the RD spectrum jointly encodes target range and radial velocity information, forming the basis for target detection, parameter estimation, and tracking. This makes weak target detection in RD spectra a core research challenge in radar signal processing [4,5].

The CFAR detector is the most mature and widely adopted conventional method for radar target detection [6]. Based on the statistical distribution of the clutter background,



Academic Editor: Jingwei Xu

Received: 26 March 2026

Revised: 5 June 2026

Accepted: 16 June 2026

Published: 3 July 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and

conditions of the [Creative Commons](https://creativecommons.org/licenses/by/4.0/)

[Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

CFAR algorithms estimate clutter power using reference cells within a sliding window, adaptively compute detection thresholds, and perform pixel-wise target/background classification while maintaining a constant system false alarm rate [7]. Since the introduction of the cell-averaging CFAR (CA-CFAR) algorithm, extensive research has been conducted to improve clutter adaptation and robustness in nonhomogeneous environments. Representative variants include the smallest-of CFAR (SO-CFAR) and greatest-of CFAR (GO-CFAR), which mitigate target masking in multi-target scenarios and threshold bias at clutter edges by taking extreme values of statistical estimates from bilateral reference windows [8]. The ordered statistic CFAR (OS-CFAR) estimates clutter power by sorting reference cell amplitudes and selecting a specific order statistic, significantly enhancing robustness in nonhomogeneous clutter [9].

In recent years, driven by demands for weak target detection, researchers have extensively explored clutter robustness, multi-target anti-interference capability, and low-SNR performance of CFAR detectors, achieving a series of incremental improvements [10–14]. Nevertheless, these methods exhibit stable detection performance only under moderate-to-high SNR. In practical low-SNR scenarios, the echo amplitude of weak targets is comparable to clutter and noise, causing the target energy to merge with the background.

Benefiting from breakthroughs in deep learning for computer vision, powerful non-linear feature extraction and data fitting capabilities have opened a new technical route for low-SNR weak target detection in radar RD spectra, offering stronger robustness than traditional CFAR detectors [4,15]. Existing deep learning-based weak target detection algorithms can be broadly categorized into two paradigms: global single-stage detectors inspired by YOLO-series frameworks [16–19], and pixel-level sliding-window classifiers based on deep convolutional neural networks (DCNNs) [20–24].

Single-stage detectors treat the RD spectrum as a 2D image and directly regress target bounding boxes and confidence scores via end-to-end global feature extraction and multi-scale fusion, avoiding pixel-wise sliding and thus achieving higher efficiency than conventional sliding-window methods. For instance, Zhao et al. proposed BCA-DetNet, which integrates a balanced channel attention module and multi-scale feature fusion to enhance low-SNR target perception, achieving over 95% detection probability at -10 dB SNR [18]. However, these methods suffer from the inherent grid quantization artifact of YOLO-style frameworks, which limits localization precision to grid resolution and degrades detection performance at -20 dB SNR. Meanwhile, global feature extraction frequently misclassifies strong clutter as targets, resulting in insufficient false alarm control. He et al. presented NST-YOLO, which uses neural style transfer for simulation-to-real domain adaptation and optimized the detection head and anchor design for small targets [19]. Nevertheless, its localization error for weak targets typically exceeds two pixels, and the false alarm rate remains above 15% under heavy clutter.

Pixel-level DCNN sliding-window methods inherit the CFAR sliding-window concept. For each pixel in the RD spectrum, a local patch centered at that pixel is extracted, and a DCNN performs target/background binary classification to enable pixel-accurate detection. Wang et al. first applied a fully convolutional DCNN to weak target detection in radar RD spectra, extracting deep features via multi-layer convolution and improving detection probability by more than 20% over CA-CFAR at -15 dB SNR [20]. However, this approach requires exhaustive global sliding-window processing, leading to computational complexity that grows quadratically with image size. To address efficiency bottlenecks, Tian et al. introduced lightweight improvements using depthwise separable convolutions and optimized sliding stride to reduce redundant computation, accelerating detection by more than three times [21].

To overcome the limitations of existing methods for weak target detection, this paper proposes a deep learning-based progressive sub-pixel detection framework that combines the speed advantage of YOLO-style detectors with the accuracy advantage of DCNN-based methods. By integrating connected component analysis and a weighted voting strategy, the proposed method realizes fast and accurate weak target detection, yielding state-of-the-art performance among deep learning-based methods in terms of speed, detection accuracy, and localization precision.

The main contributions of this paper are threefold, clearly distinguishing between domain-specific customizations, original efficiency optimizations, and synergistic design innovations:

1. Hybrid Detection Paradigm with Adaptive RoI Mechanism

We present FastDN, a radar-domain customized single-branch detection network tailored to the fixed-scale characteristic of radar targets in RD spectra. Its core design includes a single-scale detection branch to eliminate redundant computation, average pooling to preserve low-frequency diffuse energy of weak targets, and an efficient channel attention (ECA) mechanism to enhance target-related feature channels.

2. Adaptive RoI-Based Fine-Grained Detection

We introduce an adaptive region of interest (RoI) mechanism, a domain-specific efficiency optimization distinct from conventional R-CNN-style RoI pooling. Instead of processing variable-size proposals, adaptive RoI determines the search range based on the inverse relationship between detection confidence and localization uncertainty, enabling efficient fine-grained detection without proposal generation overhead.

3. Soft-Label-Driven Connected Component Weighted Localization

We develop connected component weighted localization (CCWL), a task-specific post-processing strategy whose unique advantage lies in its synergistic integration with soft probability labels from FineDN. While weighted centroid computation is mathematically classical, this combination enables sub-pixel localization precision that is unattainable with hard 0/1 labels alone.

2. Related Work

2.1. CFAR Detection

Given the 2D image structure of the RD spectrum, 2D CFAR detectors employ a 2D sliding window to estimate the clutter statistics in both range and Doppler dimensions, enabling pixel-wise target discrimination. The design objective is to maximize detection probability while maintaining a constant false alarm rate [6]. A standard 2D CFAR structure consists of three cell types: the Cell Under Test (CUT), which is the pixel being evaluated for target presence, guard cells that surround the CUT to isolate target energy leakage and prevent target signals from contaminating the clutter power estimation, and reference cells located outside the guard cells that sample background power for threshold calculation.

The schematic diagram of the 2D CA-CFAR detection algorithm is shown in Figure 1. In the figure, A_{CUT} denotes the amplitude of the CUT, $A_{\text{RC}}(i)$ is the amplitude of the i -th reference cell, Z is the sum of reference cell amplitudes, K is the threshold factor derived from the preset false alarm rate, and KZ is the adaptive detection threshold. The final detection result is determined by the binary judgment of whether $A_{\text{CUT}} > KZ$.

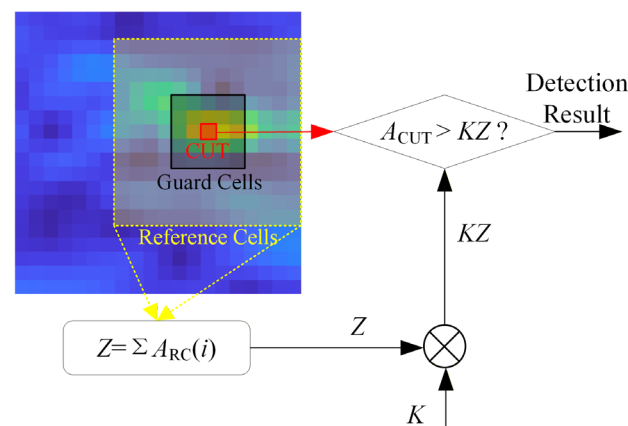


Figure 1. Schematic diagram of the 2D CA-CFAR detection algorithm.

Compared with 1D CFAR algorithms that perform clutter statistics and threshold calculation in only a single dimension, 2D CFAR exploits joint RD clutter information and provides stronger adaptability to nonhomogeneous backgrounds, establishing it as the de facto engineering benchmark for RD spectrum target detection.

2.2. YOLOv8

YOLOv8 is a state-of-the-art single-stage object detector released by Ultralytics in 2023 [25–27]. It incorporates several advanced design elements, including backbone architecture, multi-scale feature fusion [28], decoupled detection heads [29], and anchor-free paradigms [30], becoming one of the most popular frameworks in academia and industry.

The standard YOLOv8 follows a three-stage pipeline that begins with a backbone network responsible for hierarchical feature extraction from input images, ranging from shallow detailed features to deep semantic features. The backbone consists of multiple cascaded CBS modules and C2f modules, and performs four downsampling operations through convolutional layers with a stride of 2, progressively mapping the input image into multi-scale feature maps that provide inputs of different resolutions for subsequent feature fusion. The neck network adopts a bidirectional multi-scale feature fusion architecture that combines a Feature Pyramid Network (FPN) with a Path Aggregation Network (PANet). It first fuses deep, high-semantic, low-resolution features with shallow, high-resolution features through a top-down FPN pathway, and then propagates shallow detailed features to deep feature maps via a bottom-up PANet pathway, achieving bidirectional complementarity between semantic and detailed information across feature maps of different scales. The detection head employs a fully decoupled parallel branch design, separating the target classification and bounding box regression tasks into two independent convolutional branches that separately process the multi-scale feature maps output by the neck network. Additionally, YOLOv8 sets up three parallel detection heads for different scales of detection targets, corresponding to the detection requirements for large, medium, and small targets, respectively, thereby achieving differentiated feature adaptation for targets of varying scales. The network structure of YOLOv8 is illustrated in Figure 2.

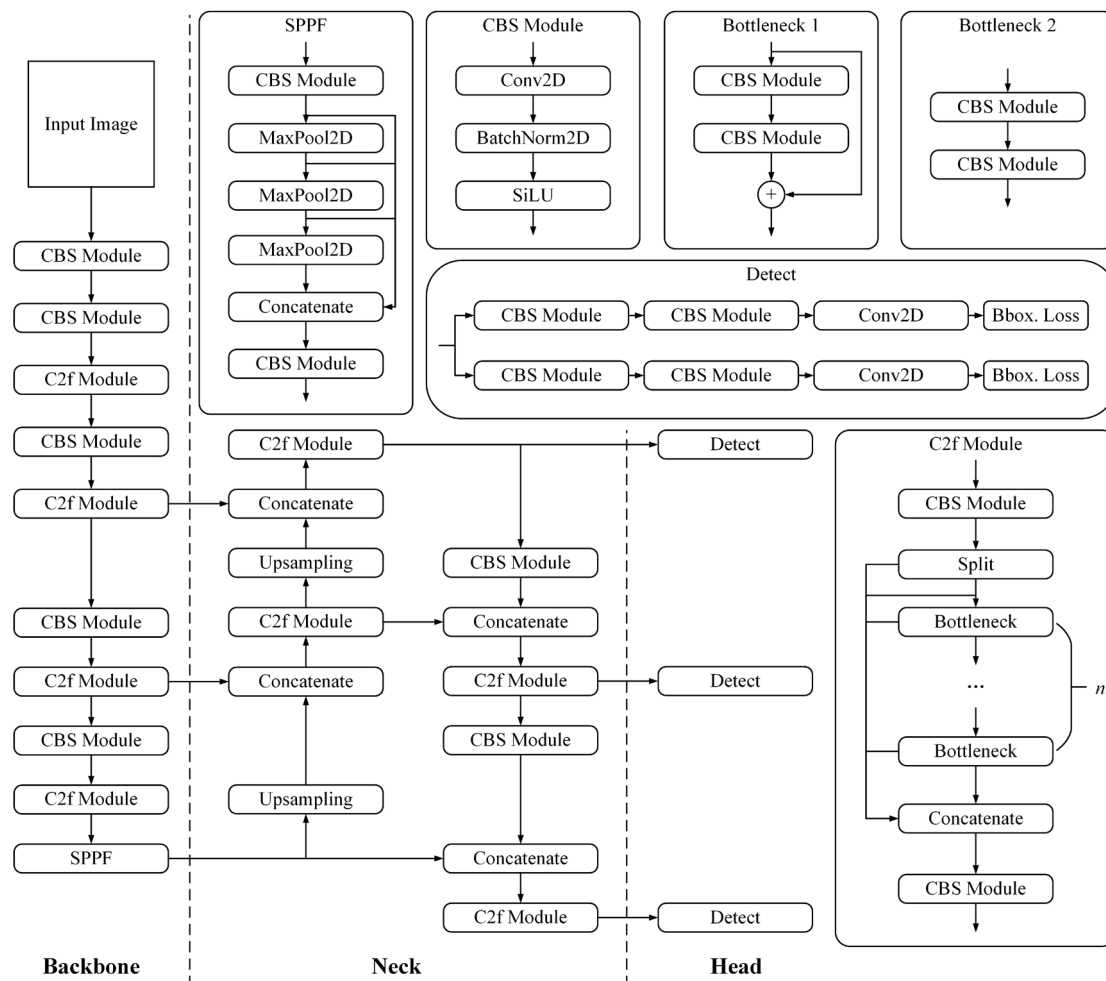


Figure 2. Architecture diagram of YOLOv8.

3. Proposed Progressive Sub-Pixel Target Detection Method

Although YOLOv8 achieves strong accuracy-speed tradeoffs in optical image object detection tasks, its native architecture suffers from significant structural mismatches when applied to weak target detection in radar RD spectra [27]. RD spectra have three core physical characteristics distinct from optical images: (1) target sparsity, with typically only 1 to 3 targets per frame; (2) fixed target scale, determined by radar resolution; and (3) weak target energy concentrated in low-frequency components, which are easily lost in standard convolution operations. Inspired by the human visual cognitive mechanism of global rapid scanning followed by local focused examination, we propose a three-stage pipeline—global fast candidate screening, local RoI fine-grained discrimination, and sub-pixel weighted localization—that leverages these characteristics to synergistically optimize of detection performance, localization accuracy, and inference efficiency.

The overall method comprises three components: Fast Detection Network (FastDN), Fine Detection Network (FineDN), and Connected Component Weighted Localization (CCWL), collectively abbreviated as F²DN-CCWL. The complete flowchart of the algorithm is shown in Figure 3.

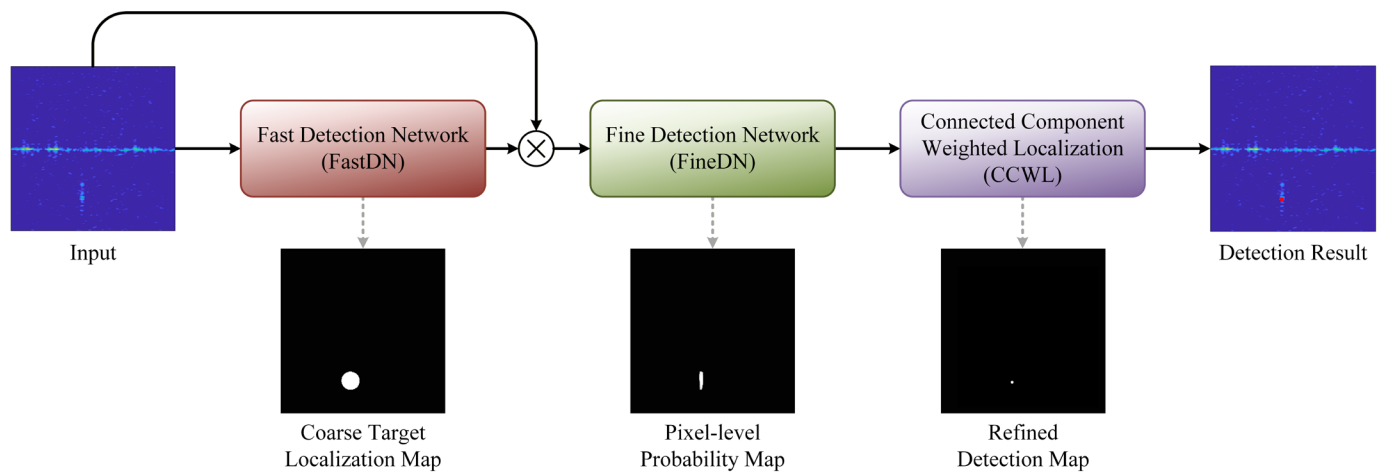


Figure 3. Overall flowchart of the proposed F²DN-CCWL algorithm. White regions in the three intermediate maps represent target-presence areas identified at successive stages from coarse localization to refined detection.

3.1. Network Input

To match the input requirements of the first-stage FastDN, the RD spectrum is resized to 288×288 pixels. This specific size is chosen because the image is divided into a 9×9 grid for fast detection, and after five convolutional downsampling operations, a 288×288 input image results in an output feature map of exactly 9×9 .

Notably, most existing methods employ moving target indication (MTI) preprocessing to suppress clutter during RD spectrum generation [20,23,31]. While MTI effectively suppresses ground clutter, it also attenuates weak target signals, increasing the risk of missed detection for targets that are already barely distinguishable from clutter in subsequent stages. The RD spectrum before and after MTI processing is shown in Figure 4, where it can be observed that although MTI suppresses the strong horizontal clutter line, it also weakens the target signal near the clutter region. In contrast, the proposed method operates directly on raw RD spectra without any clutter suppression preprocessing, which further reduces the computational burden of the overall target detection pipeline and ensures that no target of interest suffers signal loss due to preprocessing.

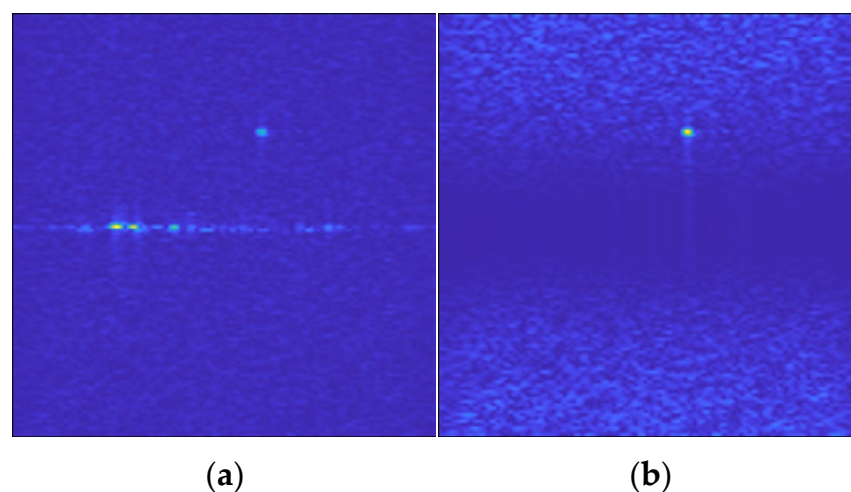


Figure 4. RD spectrum before and after MTI processing. (a) Before MTI processing; (b) After MTI processing.

3.2. Fast Detection Network (FastDN)

3.2.1. Network Structure

FastDN is a streamlined, radar-customized architecture derived from YOLOv8 architecture. When an RD amplitude image is input to the network, the backbone employs a series of convolutional neural network (CNN) layers to perform deep feature extraction. These features are then fused with multi-scale features for comprehensive target discrimination, and the detection head estimates both target presence probability and target coordinates. The overall architecture of FastDN is illustrated in Figure 5.

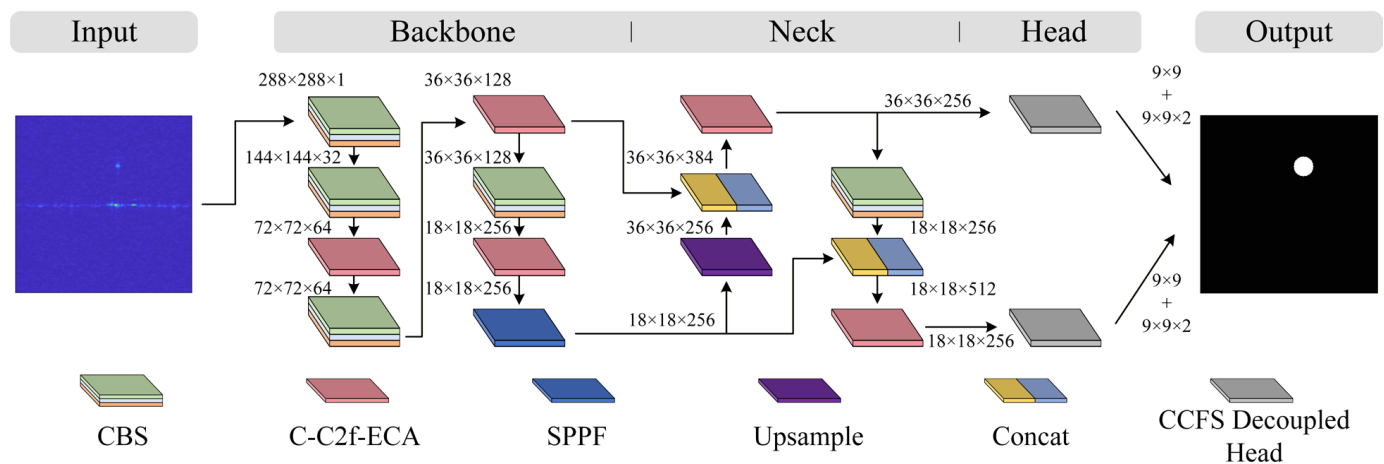


Figure 5. Architecture of the FastDN network.

Compared with YOLOv8, FastDN reduces the number of upsampling operations in the neck from two to one. This modification is motivated by the fact that, unlike optical images such as photographs or SAR images, the two dimensions of an RD spectrum do not correspond to actual physical spatial dimensions but rather the target's scattering characteristics in the range dimension and its radial velocity distribution. Consequently, the spatial extents occupied by large targets and small/weak targets in an RD spectrum typically do not differ significantly. Eliminating one scale of detection branches in the network does not noticeably degrade detection performance, but it significantly reduces the number of network parameters and computational complexity.

To better preserve the characteristics of small and weak targets in RD spectra, FastDN introduces an enhanced C2f module called the C-C2f-ECA module, whose structure is shown in Figure 6. The C-C2f-ECA module first applies a convolution operation on the input feature map and then evenly splits it evenly into two parts along the channel dimension. One part is directly convolved using a CBS module, while the other part undergoes average pooling before convolution. The two sub-feature maps with different detail levels are then concatenated along the channel dimension. Additionally, the C-C2f-ECA module incorporates the ECA module [32] to adaptively enhance target-related feature channels. Unlike SENet [33], ECA captures local cross-channel interactions without dimensionality reduction through 1D convolution, whose kernel size k is adaptively determined from the channel dimension C using Equation (1), and the resulting attention weights are applied to the input feature map (Figure 7).

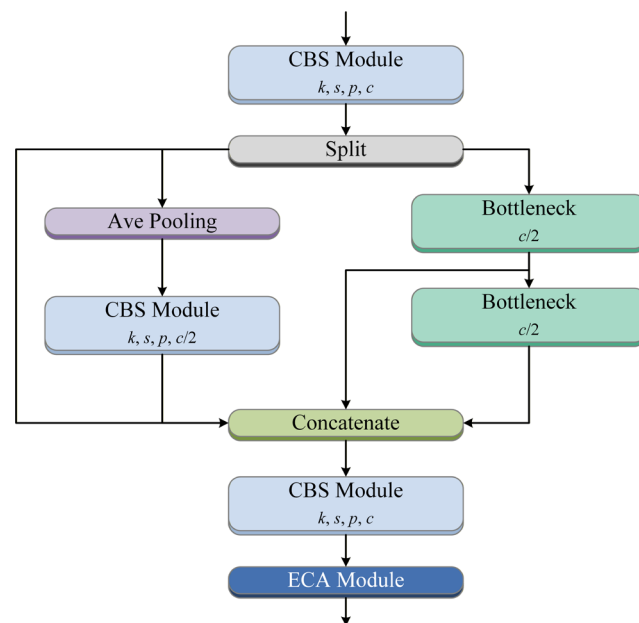


Figure 6. The structure of C-C2f-ECA.

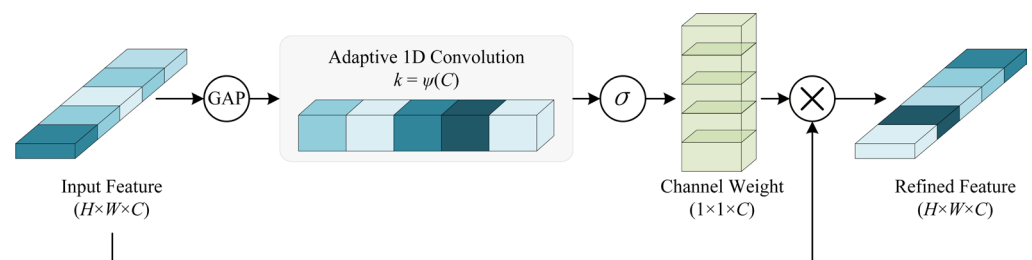


Figure 7. Workflow diagram of the ECA module.

The ECA module first applies global average pooling (GAP) to the input feature map of size $H \times W \times C$ to aggregate global spatial information. Then, a 1D convolution with kernel size k is used to generate channel attention weights, where the hyperparameter k representing the kernel size of the 1D convolution is adaptively mapped from the channel dimension C . Subsequently, the resulting feature map is passed through a Sigmoid activation function to obtain a channel attention weight vector of size $1 \times 1 \times C$, which is then multiplied element-wise with the original feature map along the channel dimension to produce an output weighted feature map of size $H \times W \times C$. The hyperparameter k can be determined by the following formula:

$$k = \psi(C) = \max\left(\left\lceil \frac{\log_2(C)}{\gamma} + \frac{\beta}{\gamma} \right\rceil_{\text{odd}}, 3\right), \quad (1)$$

where C represents the channel dimension, γ and β are hyperparameters (typically set to $\gamma = 2, \beta = 1$), and $\lceil a \rceil_{\text{odd}}$ denotes the ceiling operation to the nearest odd integer greater than or equal to a .

Therefore, the C-C2f-ECA module combines feature maps with different levels of detail through dual pathways, enriching feature combination patterns and enhancing information interaction. By leveraging channel attention to adjust the weights of each channel in the output feature map, the module ensures that fine-grained details are effectively preserved throughout the feature extraction process.

The detection head employs a decoupled architecture similar to that of YOLOv8, but replaces the final Conv2D layer with a fully connected layer and redefines the output

semantics of the network to accommodate the requirements of subsequent fine-grained detection. The structure of the CCFS decoupled head is illustrated in Figure 8.

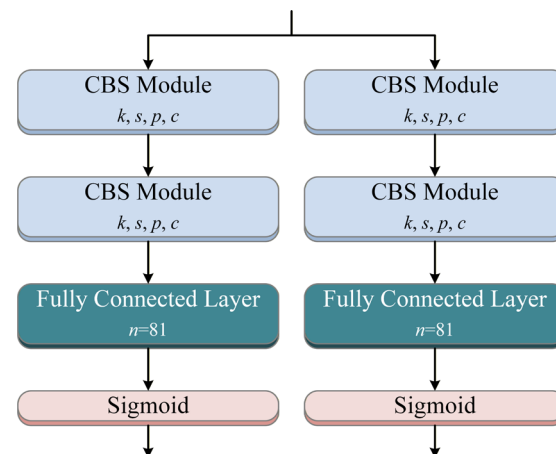


Figure 8. The structure of the CCFS decoupled head.

The first output branch predicts the probability of a target being present in each of the partitioned 9×9 grids, represented by a value between 0 and 1, resulting in a total of 81 output values. The second output branch is used to roughly estimate the normalized position of the target within its corresponding grid relative to the top-left corner of that grid, and the normalized position of each target is represented by a pair of 2D coordinates, resulting in an output of $81 \times 2 = 162$ values. The normalized coordinates of targets in the output results are shown in Figure 9. It should be noted that FastDN employs a dual decoupled head architecture. The first head operates on the 36×36 feature map to capture fine-grained positional information, while the second head operates on the 18×18 feature map after an additional CBS + C-C2f-ECA block, providing a larger receptive field beneficial for detecting weak targets. Detection results from both heads are fused using the CCWL strategy.

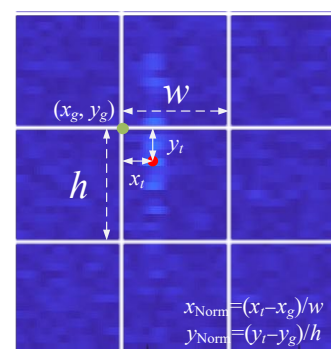


Figure 9. Illustration of normalized target coordinates within a grid cell. The green dot marks the top-left corner of the grid, and the red dot represents the ground-truth position of the target.

3.2.2. Loss Function

Considering that the output of the FastDN includes both the probability of a target being present within a grid and the normalized coordinates of the target within that grid, the proposed loss function $\text{Loss}(\cdot)$ consists of two weighted components: the target existence classification loss L_{exi} and the coordinate regression loss L_{pos} . Assuming that in a given RD spectrum, the probability that a target lies within the m -th grid is p_m , and the true normalized coordinates within the grid are $C_m = [x_m, y_m]^T$, while the corresponding

predictions from FastDN are $\hat{p}_m$ and $\hat{C}_m$, the loss function defined as the weighted sum of the existence loss and the position loss weighted by a coefficient λ :

$$\text{Loss}\left(\{C_m\}, \{p_m\}, \{\hat{C}_m\}, \{\hat{p}_m\}\right) = L_{\text{exi}}\left(\{p_m\}, \{\hat{p}_m\}\right) + \lambda \cdot L_{\text{pos}}\left(\{C_m\}, \{\hat{C}_m\}\right). \quad (2)$$

In this paper, λ is set to 81 to balance the contribution of L_{exi} and L_{pos} to network training. The existence loss L_{exi} is calculated as the sum of squared errors between the predicted and true target existence probabilities across all grids, while the position loss L_{pos} is calculated as the sum of Euclidean distances between the predicted and true coordinates only for grids that actually contain targets, which can be expressed as:

$$L_{\text{exi}}\left(\{p_m\}, \{\hat{p}_m\}\right) = \sum_{m=1}^{81} \left(p_m - \hat{p}_m\right)^2, \text{ and} \quad (3)$$

$$L_{\text{pos}}\left(\{C_m\}, \{\hat{C}_m\}\right) = \sum_{m=1}^{81} 1_m \left\|C_m - \hat{C}_m\right\|_2, \quad (4)$$

where 1_m indicates whether a target exists in the m -th grid, and $\|\cdot\|_2$ denotes the Euclidean norm.

3.3. Fine Detection Network (FineDN)

As the second stage of the target detection method, FineDN employs a DCNN architecture designed for high detection accuracy. However, unlike existing DCNN-based target detection algorithms that require exhaustive global sliding-window processing over the entire RD spectrum, the proposed method leverages the detection results from the first-stage FastDN to guide the fine-grained verification process. Specifically, only the RoIs identified by FastDN as potentially containing targets are subjected to pixel-wise verification, which drastically reduces the computational burden while maintaining high detection accuracy.

3.3.1. Adaptive RoI Determination Strategy

Since FastDN's output includes not only the estimated coordinates of potential targets but also the corresponding confidence scores representing target existence probabilities, FineDN's classification can be focused on specific regions, enabling fast target detection without global searching. A higher confidence score indicates greater similarity between the local patch centered at the estimated target coordinates and the actual target image, and vice versa. Therefore, in FineDN, the target existence confidence is used to guide the scope of fine-grained detection.

Assuming that in the output of FastDN, the target existence probability for the k -th grid is p_k , and the normalized coordinates of the target are (x_k, y_k) , the RoI search range in terms of X and Y coordinates for FineDN can be defined as:

$$\sqrt{(X - x_k)^2 + (Y - y_k)^2} \leq \frac{\Delta R}{2} \times (1 - p_k). \quad (5)$$

That is, the obtained RoI is a circular region centered at (x_k, y_k) with a radius of $\Delta R/2 \times (1 - p_k)$, and the default value of ΔR is set to 32 pixels, which exactly matches the size of each grid cell in the 9×9 partition of the 288×288 input image. This value ensures that the RoI always contains the true target even under the worst-case localization error of FastDN, while minimizing unnecessary computation. Therefore, as the target existence probability p_k output by FastDN is closer to 1, the RoI range for FineDN's fine-grained search decreases. When the target existence probability for a grid output by FastDN exceeds

0.9, the number of pixels contained in the RoI is even less than 10, which significantly reduces the time required for FineDN to perform pixel-wise target detection.

Unlike R-CNN, which generates thousands of variable-size proposals and applies feature-normalizing RoI pooling, our method directly uses FastDN's grid-level confidence outputs to define dynamic search ranges, eliminating both proposal generation overhead and feature quantization loss. This design is tailored to the sparse and fixed-scale nature of targets in RD spectra, achieving superior efficiency and information preservation.

3.3.2. Network Structure

The network structure of FineDN is relatively simple, primarily using CBS modules for feature extraction and utilizing convolutional layers with a stride of 2 for downsampling. For each pixel in the RoI, a 21×21 amplitude patch centered on that pixel is extracted as input. Although FineDN requires pixel-wise windowing for target detection, guided by the RoI, the total number of computations is reduced to approximately one-thousandth of that required by other DCNN-based target detection methods that perform detection on the entire image, achieving an excellent trade-off between detection speed and accuracy. The structure of the FineDN is illustrated in Figure 10.

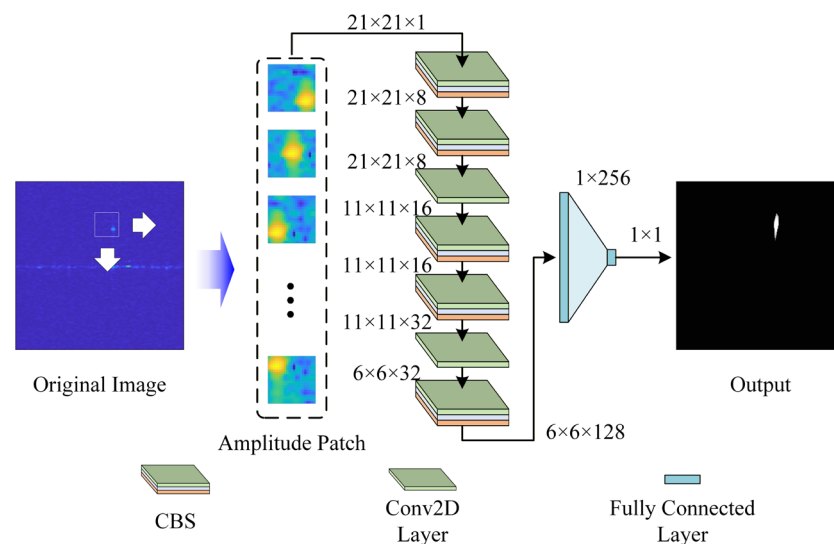


Figure 10. Architecture of FineDN.

3.3.3. Loss Function

During the training of FineDN, we propose a distance-inversely weighted soft-label ground truth to replace the conventional 0/1 binary hard label, which is highly compatible with the signal characteristics of weak targets in RD spectra. Conventional hard-label strategies assign a value of 1 exclusively at the exact target coordinate and 0 elsewhere, creating an artificially sharp decision boundary that is incompatible with the continuous physical nature of radar target responses. This forces the network to suppress physically meaningful adjacent responses, leading to slow convergence and loss of sub-pixel localization information.

To address this limitation, our soft-label ground truth assigns continuously decreasing values to pixels based on their Euclidean distance from the true target coordinate $C_t = (x_t, y_t)$. The farther the distance, the smaller the assigned value, and pixels beyond a predefined radius are set to 0. The soft-label ground truth is mathematically defined as:

$$T(C_i) = \begin{cases} 1 - \mu \|C_i - C_t\|_2, & \|C_i - C_t\|_2 \leq \frac{1}{\mu} \\ 0, & \|C_i - C_t\|_2 > \frac{1}{\mu} \end{cases}, \quad (6)$$

where C_i represents the coordinate of the current pixel, and μ is a scaling coefficient, set to 0.1 in this paper. This value is selected such that the soft-label response region approximately matches the 21×21 input patch size of FineDN, ensuring that the network learns the complete spatial distribution of target responses.

This soft-label design enables the network to learn the continuous probability distribution of target presence rather than a discrete binary classification task. It not only accelerates network convergence by providing more informative training signals but also preserves the fine-grained spatial information required for high-precision localization. Since the network output corresponds to a continuous probability response map rather than discrete detection points, the loss function is defined as the mean squared error (MSE) loss between the predicted response map and the soft-label ground truth, expressed as:

$$\text{Loss}_{\text{MSE}}(\mathbf{P}, \mathbf{T}) = \frac{1}{WH} \sum_{y=1}^W \sum_{x=1}^H (p(x, y) - T(x, y))^2, \quad (7)$$

where $\mathbf{P} = \{P(x, y) | x \in \{1, \dots, W\}, y \in \{1, \dots, H\}\}$ represents the prediction map output by the network, and W and H denote the width and height of the image, respectively. The continuous probability outputs generated by this soft-label training strategy form the essential foundation for the subsequent CCWL post-processing module, enabling the sub-pixel-level localization accuracy that cannot be achieved with conventional hard-label-based detection methods.

Based on the previously detailed structures of FastDN and FineDN, the complete architecture of the F²DN network is summarized in Table 1.

Table 1. The Complete Architecture of the F²DN Network.

Module	Kernel Size	Number of Kernels/Neurons	Stride	Activation Function	Output Size
CBS	3×3	32	2	SiLU	$144 \times 144 \times 32$
CBS	3×3	64	2	SiLU	$72 \times 72 \times 64$
C-C2f-ECA	3×3	64	1	SiLU	$72 \times 72 \times 64$
CBS	3×3	128	2	SiLU	$36 \times 36 \times 128$
C-C2f-ECA	3×3	128	1	SiLU	$36 \times 36 \times 128$
CBS	3×3	256	2	SiLU	$18 \times 18 \times 256$
C-C2f-ECA	3×3	64	1	SiLU	$18 \times 18 \times 256$
SPPF	1×1	256	1	SiLU	$18 \times 18 \times 256$
Upsampling Layer	-	-	2	-	$36 \times 36 \times 256$
C-C2f-ECA	3×3	256	1	SiLU	$36 \times 36 \times 256$
CCFS Decoupled Head	3×3	81	1	SiLU/Sigmoid	1×81 (Reshape to 9×9)
CBS	3×3	162	1	SiLU/Sigmoid	1×162 (Reshape to $9 \times 9 \times 2$)
CBS	3×3	256	2	SiLU	$18 \times 18 \times 256$
C-C2f-ECA	3×3	64	1	SiLU	$18 \times 18 \times 256$
CCFS Decoupled Head	3×3	81	1	SiLU/Sigmoid	1×81 (Reshape to 9×9)
Head	3×3	162	1	SiLU/Sigmoid	1×162 (Reshape to $9 \times 9 \times 2$)

FastDN employs dual CCFS Decoupled Heads: first head on 36×36 feature map for fine positional information, second head on 18×18 feature map (after additional C-C2f-ECA) for larger receptive field beneficial to weak targets.

3.4. Network Training

The FastDN and FineDN networks are trained separately for parameter optimization and are combined after completing their respective training. Therefore, it is necessary to construct training data and ground truth labels for each network individually.

As discussed in Section 3.1, we omit MTI preprocessing to preserve weak target signals, but this introduces the risk of the network overfitting to spatially concentrated ground

clutter. To mitigate this risk, we employ a random 2D circular shift augmentation strategy on the original RD spectra during FastDN training to eliminate the spatial dependence of clutter distribution.

The positions of clutter and targets in the RD spectra before and after random 2D circular shifting are illustrated in Figure 11. This augmentation ensures that clutter is uniformly distributed across different grid partitions, enabling each grid region to achieve a balanced classification performance with respect to targets, clutter, and noise after training.

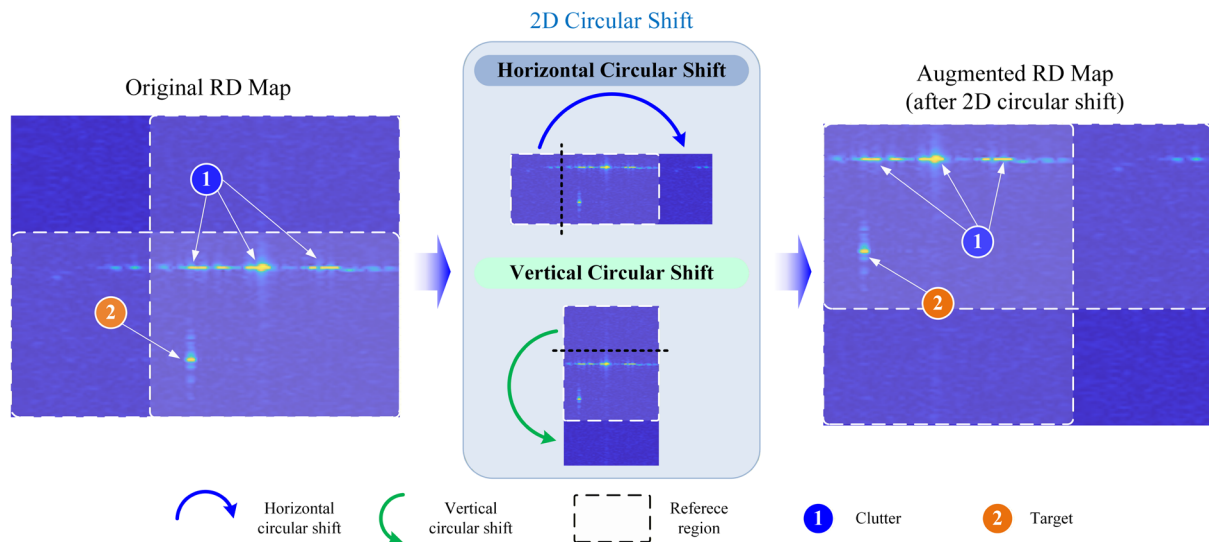


Figure 11. Distribution of targets and clutter in the RD spectrum before and after random 2D circular shift augmentation. The black dotted lines indicate the split boundaries of the RD spectrum along the two dimensions during the 2D cyclic shift.

Since each CCFS decoupled head in FastDN contains two outputs, where one represents the probability of a target existing in the grid and the other represents the normalized coordinates of the target within the grid, the ground truth for FastDN training consists of these two corresponding components. The first part of the ground truth is constructed by uniformly dividing the RD spectrum into a 9×9 grid. Based on the target coordinates, the grid containing the target is labeled as 1, while all other grids are labeled as 0. The second part of the ground truth is derived from the grids labeled as 1 in the first part. For each grid, the normalized coordinates corresponding to the target it contains are calculated as the ground truth, while all grids without targets are labeled as (0,0).

During training, the original samples (directly generated radar RD spectra) and the augmented samples obtained through 2D circular shifting are thoroughly mixed and shuffled in a 1:3 ratio, and then fed into the network in mini-batches. The training hyperparameters for the two networks are presented in Table 2.

Table 2. F²DN Training Parameters.

Network Name	Hyperparameter	Value
FastDN	Optimizer	SGDM
	Momentum	0.9
	Initial Learning Rate	0.02
	Learn Rate Drop Factor	0.2
	Learn Rate Drop Period	5
	Epochs	20

Table 2. *Cont.*

Network Name	Hyperparameter	Value
FastDN	Batch Size	64
	Sample SNR (dB)	−20 to 0
	Number of Samples	Original Samples: 500 Augmented Samples: 1500
FineDN	Optimizer	SGDM
	Momentum	0.9
	Learning Rate (constant)	0.01
	Epochs	50
	Batch Size	64
	Sample SNR (dB)	−20 to 0
	Number of Samples	7500

3.5. Weighted Localization Algorithm

After the aforementioned network processing, the final output assigns a value between 0 (no target) and 1 (target present) to each detection position. This value represents the similarity between the 21×21 image patch centered at the current detection pixel and the RD spectrum characteristic of a target. Therefore, when this value exceeds 0.5, we consider it more than 50% probable that a target exists at the current position. This threshold corresponds to the midpoint of the soft-label decay function (6), providing a physically motivated decision boundary. Its optimality was further confirmed on the validation set in Section 4.6. All outputs at positions below this threshold are set to zero.

However, because an image patch remains highly similar to the original after a small translation, DCNN-based algorithms often produce multiple detection results exceeding the threshold around the true target position, rather than only one or a few positions as is typical with CFAR detection algorithms, as is typically the case with CFAR detection algorithms. This results in many methods achieving a high target detection probability but also a high false detection probability. Some algorithms employ non-maximum suppression (NMS) to retain only the maximum response value. However, this approach strictly limits the detected target position and velocity to the resolution of the RD spectrum. To obtain more accurate target position and velocity estimates, we integrate connected component analysis from image processing with the proposed target detection method, introducing CCWL, a task-specific post-processing strategy. The CCWL method consists of the following three steps:

1. Connected Component Extraction

In the feature image after threshold decision, detect the number of non-zero regions and extract all connected components as candidate targets. This process is implemented using depth-first search (DFS). The core logic is as follows: first, extract the coordinates of all non-zero pixels in the image; traverse each non-zero pixel. If a pixel has not been marked (i.e., its value has not been set to zero), it is treated as the starting point of a new connected component. Recursively traverse all non-zero pixels in its 8-neighborhood, setting the traversed pixels to zero to mark them as processed, until all pixels of the connected component are labeled. Finally, the number of starting points is counted as the total number of non-zero connected components. The pseudocode for connected component extraction is shown in Algorithm 1.

Algorithm 1: Connected Component Extraction Based on DFS

Input: Thresholded feature image I (size: $H \times W$, where non-zero pixels represent potential target responses)

Output: List of connected components C (each element is a set of pixel coordinates belonging to one connected component), Total number of connected components N_{CC}

Initialize an empty list $C \leftarrow \emptyset$ to store all connected components

Get the dimensions of the input image: $H \leftarrow \text{Height}(I)$, $W \leftarrow \text{Width}(I)$

for each pixel coordinate (x, y) in I **do**

if $I(x, y) > 0$ **then**

 Initialize an empty set $C_{cur} \leftarrow \emptyset$

 Call $DFS(x, y, C_{cur}, I, H, W)$

 Add current_component to C : $C \leftarrow C \cup C_{cur}$

end if

end for

Calculate the total number of connected components: $NCC \leftarrow \text{length}(C)$

return C, N_{CC}

Sub-function: $DFS(x, y, C_{cur}, I, H, W)$

Add the current pixel coordinate to the C_{cur} : $C_{cur} \leftarrow C_{cur} \cup \{(x, y)\}$

Mark the current pixel as processed by setting its value to 0: $I(x, y) \leftarrow 0$

for each direction offset $(dx, dy) \in \{(-1, -1), (-1, 0), (-1, 1), (0, -1), (0, 1), (1, -1), (1, 0), (1, 1)\}$ **do**

 Calculate the neighbor coordinate: $x' \leftarrow x + dx$, $y' \leftarrow y + dy$

if $1 \leq x' \leq H$ and $1 \leq y' \leq W$ (neighbor is within image bounds) **and** $I(x', y') > 0$ (neighbor is unprocessed) **then**

 Recursively call $DFS(x', y', C_{cur}, I, H, W)$

end if

end for

2. False Alarm Region Suppression

To avoid false alarms caused by erroneous judgments of sparse noise, during connected component analysis, if the number of pixels within a connected component is less than 20, the region is considered to be the result of false detections and is discarded.

This threshold is rigorously derived from our soft-label training strategy (Equation (6)). For a detection threshold of 0.5, the theoretical target response region has a radius of 5 pixels (≈ 81 pixels total). We set $T_{cc} = 20$ pixels to provide a safety margin against partial response attenuation while effectively filtering isolated noise spikes.

3. Target Position Weighted Voting

After the above steps, the connected component corresponding to the final detected target is obtained. Each pixel within the connected component has a value representing the probability that the pixel location corresponds to the true target position. As shown in Figure 12, matrix values are the target existence confidence of each pixel, where 0 indicates no target, and values closer to 1 indicates higher probability of the pixel being the real target position.

In order to obtain the final detection result, a weighted summation is performed on the pixels within each connected component. Assume that the connected component to be weighted and voted on is denoted as Z . First, sum the values of all pixels in Z to obtain the total weight W :

$$W = \sum_{p \in Z} v_i. \quad (8)$$

Next, divide the value of each pixel by the total weight W to obtain the weight w_i for that pixel position:

$$w_i = \frac{v_i}{W}. \quad (9)$$

Finally, multiply the x -coordinate x_i and y -coordinate y_i of each pixel in the connected component by its respective weight proportion w_i and sum the results to obtain the final target detection coordinates x_{ave} and y_{ave} :

$$x_{ave} = \sum_{p \in Z} x_i w_i, \text{ and} \quad (10)$$

$$y_{ave} = \sum_{p \in Z} y_i w_i. \quad (11)$$

After processing with the CCWL method, a unique set of coordinates is obtained for each connected component, which represents the final target detection result.

0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.7216	0.8153	0.8782	0.8104	0.5774	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.7453	0.8460	0.8868	0.8016	0.5448	0.0000	0.0000
0.0000	0.0000	0.0000	0.7431	0.9236	0.9518	0.9621	0.9219	0.7549	0.0000	0.0000
0.0000	0.0000	0.0000	0.7360	0.9395	0.9578	0.9764	0.9459	0.8542	0.5059	0.0000
0.0000	0.0000	0.5731	0.8567	0.9779	0.9863	0.9911	0.9765	0.9022	0.5939	0.0000
0.0000	0.0000	0.5490	0.8650	0.9772	0.9885	0.9937	0.9842	0.9351	0.6012	0.0000
0.0000	0.0000	0.6503	0.9164	0.9894	0.9940	0.9961	0.9915	0.9464	0.6158	0.0000
0.0000	0.0000	0.0000	0.8325	0.9757	0.9885	0.9942	0.9862	0.9237	0.5277	0.0000
0.0000	0.0000	0.6439	0.8897	0.9779	0.9868	0.9914	0.9815	0.9074	0.0000	0.0000
0.0000	0.0000	0.6146	0.8619	0.9657	0.9692	0.9782	0.9530	0.8182	0.0000	0.0000
0.0000	0.0000	0.7123	0.9074	0.9715	0.9746	0.9722	0.9451	0.7798	0.0000	0.0000
0.0000	0.0000	0.7996	0.9520	0.9774	0.9753	0.9640	0.9068	0.6931	0.0000	0.0000
0.0000	0.5439	0.8589	0.9498	0.9763	0.9805	0.9644	0.9203	0.7122	0.0000	0.0000
0.0000	0.6726	0.9014	0.9742	0.9909	0.9904	0.9875	0.9507	0.7792	0.0000	0.0000
0.0000	0.6896	0.9333	0.9831	0.9932	0.9938	0.9903	0.9704	0.8380	0.0000	0.0000
0.0000	0.7204	0.9565	0.9927	0.9978	0.9979	0.9973	0.9888	0.9219	0.0000	0.0000
0.0000	0.6777	0.9584	0.9944	0.9982	0.9987	0.9975	0.9907	0.9387	0.0000	0.0000
0.0000	0.5539	0.9291	0.9917	0.9974	0.9982	0.9976	0.9885	0.9232	0.0000	0.0000
0.0000	0.0000	0.9039	0.9904	0.9976	0.9985	0.9976	0.9909	0.9176	0.0000	0.0000
0.0000	0.0000	0.7846	0.9684	0.9908	0.9943	0.9912	0.9654	0.7893	0.0000	0.0000
0.0000	0.0000	0.7529	0.9598	0.9875	0.9915	0.9854	0.9639	0.7550	0.0000	0.0000
0.0000	0.0000	0.7132	0.9329	0.9754	0.9755	0.9509	0.8714	0.0000	0.0000	0.0000
0.0000	0.0000	0.7335	0.9239	0.9579	0.9562	0.9046	0.7632	0.0000	0.0000	0.0000
0.0000	0.0000	0.5762	0.8353	0.9198	0.9131	0.8201	0.6093	0.0000	0.0000	0.0000
0.0000	0.0000	0.5907	0.8224	0.8842	0.8902	0.7353	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.6277	0.7947	0.7978	0.6757	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.5804	0.6695	0.7017	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.5897	0.6067	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

Figure 12. Schematic diagram of detection results for a connected component.

4. Experimental Results

4.1. Dataset

To comprehensively validate the performance of the proposed F²DN-CCWL method for weak target detection, we conducted target detection experiments using both simulated RD spectra with varying SNR and real RD spectra. The SNR of the simulated data ranged from -20 dB to 0 dB in 5 dB increments. All SNR values reported in this paper are strictly defined as the peak SNR after pulse compression, which directly corresponds to the target's energy intensity in the final RD spectrum input to our detection algorithm. This definition is adopted because it eliminates the variability introduced by different pulse compression gains across radar systems, providing a unified benchmark that directly reflects the actual detection difficulty in the RD spectrum domain.

For each SNR level, 2000 data frames were generated, with target positions randomly placed within the RD spectra. The simulated dataset contains a total of 10,000 frames (2000 frames per SNR level), split into 70% training set (7000 frames), 15% validation set (1500 frames), and 15% test set (1500 frames).

The real-world data were acquired through actual flight experiments using an X-band anti-UAV radar with a DJI Phantom 4 UAV as the target, and a total of 2000 annotated valid data frames were recorded. Measurements show that the target SNR in the real-world data is mainly distributed between -20 dB and -15 dB. The real-world dataset is split into 50% training set (1000 frames), 25% validation set (500 frames), and 25% test set (500 frames). All model selection and hyperparameter tuning were performed exclusively on the validation set, and the test set was kept completely unseen until final evaluation.

During data collection, the radar under test was installed on an open rooftop, and the UAV flew in radial round-trip trajectories approximately along the normal direction of the radar antenna array, with a maximum flight distance of 3 km. A photograph of the radar used for data collection is shown in Figure 13. The parameters of both the simulated and real-world data are presented in Table 3.



Figure 13. Photograph of the X-band radar used for data acquisition.

Table 3. Parameter of Experimental Data.

Parameter	Symbol	Value
Carrier Frequency	F	9.5 GHz
Pulse Width	τ	4 μ s
Signal Bandwidth	B	40 MHz
Number of Pulses	N	128
Pulse Repetition Frequency (PRF)	PRF	32 kHz
SNR (simulated)	SNR	−20~0 dB

4.2. Experimental Environment

All experiments were conducted on a consumer-grade computer, and the hardware and software configurations are detailed in Table 4.

Table 4. Configuration Parameters of Experimental Platform.

Item	Configuration
Operating System	Windows 11
CPU	Intel(R) Core(TM) Ultra 9 275HX (2.70 GHz)
Memory	32 GB
GPU	NVIDIA GeForce RTX 5060
Programming Language	MATLAB R2025b

4.3. Evaluation Metrics

For consistency, all SNR values presented in the experimental results and discussions refer to the peak SNR after pulse compression as defined in Section 4.1. Since deep learning-based target detection methods cannot theoretically derive the detection probability P_d under a given false alarm rate P_f in the same manner as traditional CFAR detection algorithms, we redefine P_d and P_f to form a unified evaluation metric applicable to various

detection algorithms. Assuming the test dataset contains a total of N targets, where N_t is the number of correctly detected targets, N_f is the number of falsely detected targets, and N_m is the number of missed detections, the evaluation metrics P_d and P_f used to assess the target detection performance of the algorithm can be expressed as:

$$P_d = \frac{N_t}{N} = \text{Recall}, \text{ and} \quad (12)$$

$$P_f = \frac{N_f}{(N_t + N_f)} = 1 - \text{Precision}. \quad (13)$$

It should be noted that the P_f defined in this paper is the target-level false alarm rate, which is different from the conventional cell-level CFAR false alarm rate. As can be seen from the equations defined, P_d and P_f are essentially equivalent to Recall and 1-Precision, which are commonly used in deep neural networks. Therefore, these metrics can simultaneously evaluate the performance of both deep learning-based target detection methods and traditional CFAR detection algorithms.

While receiver operating characteristic (ROC) analysis provides a comprehensive view of performance across all operating points, single-point evaluation at the optimal threshold is widely used in radar detection research. This is because practical radar systems operate at a single pre-defined threshold optimized for specific mission requirements. The threshold of $\tau = 0.5$ used in this work is both physically motivated—corresponding to the midpoint of the soft-label decay function—and confirmed on the validation set to achieve the best balance between detection probability and false alarm rate.

Furthermore, whether a target is correctly detected is inherently a subjective criterion. To enable quantitative evaluation of the metrics, we define the detection localization error d_{pos} as the Euclidean distance between the detected target coordinates $\hat{C}_d$ and the ground-truth target coordinates C_t . A threshold T_{dis} is then set to distinguish between correct detections and false detections, defined as:

$$H_1 = \begin{cases} 1, & d_{\text{pos}} \leq T_{\text{dis}}, \text{ and} \\ 0, & d_{\text{pos}} > T_{\text{dis}} \end{cases} \quad (14)$$

$$d_{\text{pos}} = \|\hat{C}_d - C_t\|_2, \quad (15)$$

where H_1 denotes the hypothesis of a correct detection; when $H_1 = 1$, the target is considered to have been correctly detected. In this paper, T_{dis} is set to 5 pixels.

To compare the localization accuracy of different algorithms across different algorithms and datasets, we also compute the minimum localization error d_{min} and the average localization error d_{ave} of the target detection results, defined as:

$$d_{\text{min}} = \min_{n=1,2,\dots,N} (d_{\text{pos},n}), \text{ and} \quad (16)$$

$$d_{\text{ave}} = \frac{1}{M} \sum_{M=1}^M d_{\text{pos},m}, \quad (17)$$

where N is the total number of targets in the dataset, and $M = N_t + N_f$ is the total number of detections made by the algorithm.

4.4. Comparative Experiments

Using the parameters listed in Table 3, we simulated UAV echo data with SNR ranging from -20 dB to 0 dB, with an interval of 5 dB between successive levels. All baseline algorithms were strictly implemented according to their original papers and retrained and

tuned on the same training and validation datasets as our proposed method to ensure a fair comparison. CA-CFAR [6] uses 2×2 guard cells, 8×8 reference cells, and the threshold factor was tuned to 12.5 on the validation set; DCNN1 [20] and DCNN2 [21] used exactly the same network architecture and training hyperparameters as in the original publications; NST-YOLO [19] and BCA-DetNet [18] were initialized with pre-trained optical image weights and then fine-tuned on our radar dataset.

The statistical results of the detection probability P_d and false alarm rate P_f for each algorithm are presented in Table 5.

Table 5. Comparisons of Detection Performance of Various Methods on Simulated Data with Different SNRs.

SNR	CA-CFAR [6]		DCNN1 [20]		DCNN2 [21]		NST-YOLO [19]		BCA-DetNet [18]		F ² DN-CCWL	
	P_d	P_f	P_d	P_f	P_d	P_f	P_d	P_f	P_d	P_f	P_d	P_f
−20 dB	0.704	0.607	0.904	0.313	0.873	0.046	0.864	0.171	0.941	0.067	0.953	0.031
−15 dB	0.789	0.578	0.896	0.299	0.885	0.042	0.879	0.153	0.948	0.066	0.961	0.033
−10 dB	0.836	0.472	0.927	0.270	0.904	0.035	0.897	0.162	0.954	0.057	0.964	0.027
−5 dB	0.890	0.290	0.940	0.269	0.945	0.036	0.928	0.171	0.960	0.051	0.975	0.022
0 dB	0.939	0.191	0.961	0.226	0.938	0.029	0.931	0.156	0.964	0.048	0.976	0.018

As shown in Table 5, all algorithms exhibit improved performance with increasing SNR. To provide a more intuitive visualization of the performance trends across the entire SNR range, we plot the P_d and P_f curves for all compared methods in Figure 14.

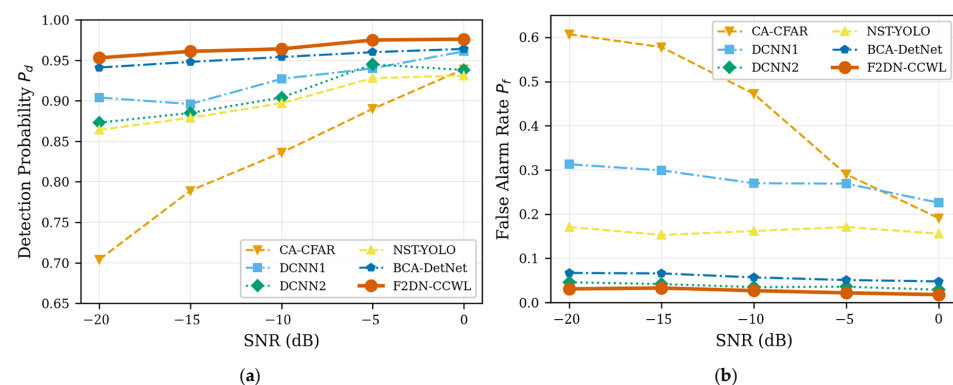


Figure 14. Performance comparison of different methods under varying SNR conditions. (a) P_d vs. SNR; (b) P_f vs. SNR.

From Figure 14a, it can be observed that F²DN-CCWL maintains the highest detection probability across all tested SNR levels. At the most challenging −20 dB condition, it achieves a P_d of 95.3%, which is 1.2 percentage points higher than the second-best BCA-DetNet [18]. As SNR increases, F²DN-CCWL demonstrates stable detection capability under varying signal conditions. In contrast, traditional CA-CFAR degrades catastrophically at low SNR, with its P_d dropping to only 70.4% at −20 dB. Figure 14b highlights the superior false alarm control capability of the proposed method. F²DN-CCWL achieves the lowest P_f at all SNR levels, with only 3.1% at −20 dB, which is 67.4% of that of DCNN2 [21] and less than one-fifth of that of NST-YOLO [19]. As SNR increases from −20 dB to 0 dB, the P_f of F²DN-CCWL steadily decreases to 1.8%, demonstrating that the progressive detection architecture and CCWL post-processing effectively suppress false alarms.

The experimental results demonstrate that the proposed progressive two-stage detection architecture effectively addresses the inherent trade-off among detection efficiency, detection accuracy, and false alarm control commonly encountered in traditional single-

stage detection networks and global sliding-window detection methods. By leveraging FastDN for rapid localization of RoI and significant compression of the search range, combined with FineDN for fine-grained sliding-window detection within local regions, the method achieves balanced performance. Furthermore, the introduction of the CCWL module further enhances the algorithm’s ability to suppress false alarms under strong clutter backgrounds.

Table 6 presents the minimum localization error $d_{\min}$ and the average localization error d_{ave} of the detection results obtained by different target detection algorithms under each SNR condition. It can be observed that the traditional CA-CFAR [6] algorithm, constrained by the statistical estimation characteristics of the constant false alarm threshold, is highly susceptible to background interference under low SNR conditions, producing a large number of outlier false alarm points and consequently leading to catastrophic degradation in localization performance. Single-stage detection algorithms based on the YOLO framework are limited by the grid quantization artifact, resulting in minimum localization errors generally higher than those of pixel-level detection methods. At an SNR of -20 dB, the minimum localization errors reach 1.37 pixels and 2.82 pixels, respectively. Moreover, the insufficient feature extraction capability for small and weak targets under low SNR conditions leads to a significant increase in false alarms, resulting in a relatively high average localization error. In contrast, the proposed F²DN-CCWL method achieves an average localization error of 0.76 pixels even under the -20 dB SNR condition, which is 65.1% lower than that of DCNN1 [20], the suboptimal method in the same scenario. Furthermore, it consistently maintains low minimum and average localization errors even under high SNR conditions, demonstrating a clear advantage in localization accuracy.

Table 6. Minimum and Average Localization Errors Obtained using Different Algorithms.

SNR	CA-CFAR [6]		DCNN1 [20]		DCNN2 [21]		NST-YOLO [19]		BCA-DetNet [18]		F ² DN-CCWL	
	$d_{\min}$	d_{ave}	$d_{\min}$	d_{ave}	$d_{\min}$	d_{ave}	$d_{\min}$	d_{ave}	$d_{\min}$	d_{ave}	$d_{\min}$	d_{ave}
-20 dB	38.93	54.14	0.19	2.18	0.18	22.51	1.37	19.86	2.82	10.84	0.04	0.76
-15 dB	0.83	42.90	0.12	2.39	0.15	19.44	0.29	18.64	2.59	8.63	0.02	0.63
-10 dB	0.00	21.38	0.05	1.78	0.13	18.76	0.47	15.53	2.32	7.49	0.01	0.72
-5 dB	0.00	15.87	0.06	1.38	0.12	15.39	0.46	8.29	2.26	5.33	0.00	0.73
0 dB	0.00	18.63	0.07	0.61	0.13	9.67	0.35	5.76	2.11	4.16	0.01	0.78

To thoroughly validate the engineering practicality and clutter robustness of the proposed F²DN-CCWL algorithm in real-world complex electromagnetic environments, a comparative experiment was conducted using 500 frames of RD spectra acquired from actual flight detection of a DJI Phantom 4 UAV using an X-band anti-UAV radar. The detection probability P_d and false alarm rate P_f of each algorithm on the measured dataset are presented in Table 7.

Table 7. The P_d and P_f for Real Data Obtained using Different Algorithms.

CA-CFAR [6]		DCNN1 [20]		DCNN2 [21]		NST-YOLO [19]		BCA-DetNet [18]		F ² DN-CCWL	
P_d	P_f	P_d	P_f	P_d	P_f	P_d	P_f	P_d	P_f	P_d	P_f
0.288	0.419	0.917	0.277	0.854	0.249	0.849	0.165	0.934	0.408	0.958	0.037

As shown in Table 7, compared with traditional algorithms, all deep learning-based detection methods still achieve superior detection performance in real-world scenarios, albeit with varying degrees of performance degradation due to clutter. In contrast, the proposed F²DN-CCWL algorithm, which employs a progressive two-stage detection architecture consisting of FastDN and FineDN, does not exhibit any significant degradation

in detection performance. Under real-world data conditions, the F²DN-CCWL algorithm achieves a detection probability of 95.8%, which is 2.4 percentage points higher than that of the BCA-DetNet [18] algorithm, while maintaining a false alarm rate of 3.7%, which is only 22.42% of that of the NST-YOLO [19] algorithm under the same scenario.

The comparable detection performance between real-world data (95.8% P_d) and simulated -20 dB data (95.3% P_d) arises from two counteracting factors: the real-world data has a higher mean SNR (approximately -12 dB) than the simulated -20 dB condition, which improves detection; however, the real-world clutter is more complex, which degrades detection. These factors partially cancel, resulting in similar performance.

Figure 15 illustrates the target detection results for selected RD spectra from the real dataset. Figure 15a shows the input RD spectrum to FastDN; Figure 15b shows the FastDN output, which provides coarse target localization; Figure 15c shows the output of FineDN, which presents pixel-level probability responses with multiple neighboring peaks around the true target; Figure 15d shows the final CCWL output, which achieves accurate sub-pixel localization with all false alarms suppressed. This figure clearly demonstrates the complementary roles of each stage in our pipeline.

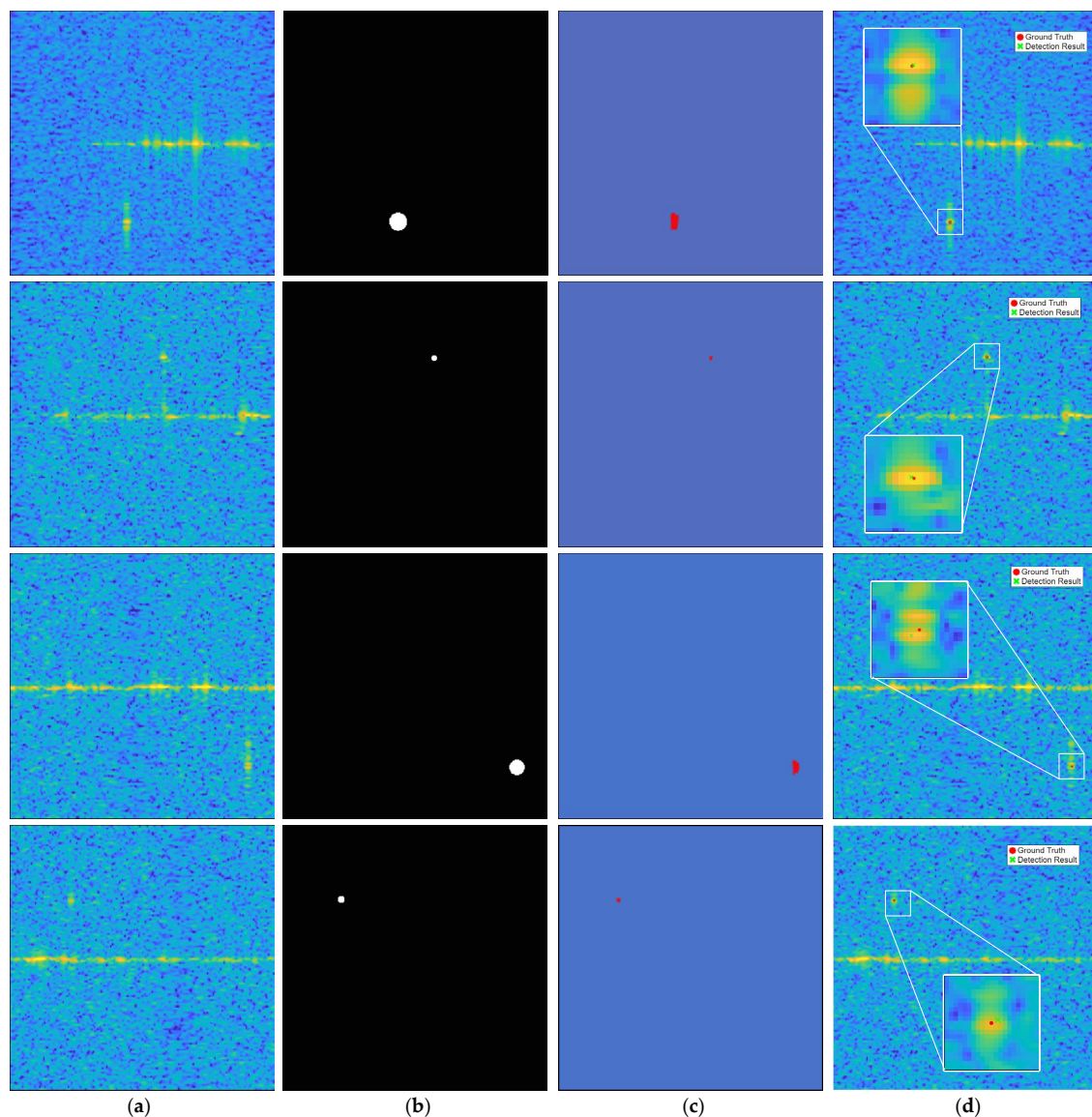


Figure 15. Target detection results of sample RD spectra. (a) Input RD spectra; (b) FastDN output; (c) FineDN output; (d) Final detection results.

We also evaluated multi-target detection performance on simulated RD spectra containing 2 to 3 randomly distributed targets at -20 dB SNR. These represent the most common scenarios in practical anti-UAV applications. The results in Table 8 show that our method maintains a detection probability of over 95% while keeping the false alarm rate below 4%, outperforming all baselines by a significant margin.

Table 8. Multi-Target Detection Performance at -20 dB SNR.

Number of Targets	Metric	CA-CFAR [6]	DCNN1 [20]	DCNN2 [21]	NST-YOLO [19]	BCA-DetNet [18]	F ² DN-CCWL
2	P_d	0.710	0.897	0.862	0.848	0.946	0.949
	P_f	0.621	0.306	0.044	0.175	0.073	0.031
3	P_d	0.698	0.901	0.869	0.851	0.944	0.951
	P_f	0.614	0.322	0.047	0.177	0.072	0.037

4.5. Ablation Experiment

To comprehensively validate the effectiveness, performance contribution, and functional complementarity of the three core modules in the proposed F²DN-CCWL target detection method, we conducted ablation experiments by comparing four different configurations that incorporate one or two of the components of FastDN, FineDN, and CCWL against the complete F²DN-CCWL method. The detection performance, localization accuracy, and computational efficiency of the individual modules, pairwise combinations, and the full algorithm were evaluated. The experimental results are summarized in Table 9.

The ablation experiment results show that FastDN serves as the core enabler for achieving real-time and efficient detection. Experimental Group A, which uses only FastDN, achieves a detection probability of 97.2% with a processing time of only 12.96 ms per frame, fully verifying that this module, through global fast feature extraction and grid-level target pre-detection, can achieve high-recall global screening of weak targets with very low computational overhead. However, due to the grid quantization artifact inherent to the YOLO architecture, the false alarm rate of FastDN alone reaches 35.6%. Compared with Group A, Experimental Group B, which only uses FineDN, achieves significantly improved localization accuracy, with the minimum localization error reduced to 0.21 pixels, verifying that the pixel-wise sliding-window detection mode of this module enables finer target feature extraction and pixel-level discrimination. However, FineDN alone requires global traversal detection across the entire RD spectrum, resulting in prohibitive computational overhead (3051.35 ms per frame), making it impractical for real-time engineering applications. It should be noted that the extremely high false alarm rates in Groups B and C arise from the fact that a detection threshold of 0.5 was still used during the ablation experiments, while FineDN tends to generate responses exceeding the threshold within a certain range around the true target, resulting in an excessively high false alarm rate.

Table 9. Results of Ablation Experiments on Real-World Dataset.

Experiment Number	FastDN	FineDN	CCWL	P_d	P_f	$d_{\min}$	d_{ave}	Time Consumption (ms)
A	✓	×	×	0.972	0.456	2.63	17.53	12.96
B	×	✓	×	0.986	0.972	0.00	9.24	3051.35
C	×	✓	✓	0.983	0.129	0.05	1.86	3062.69
D	✓	✓	×	0.961	0.957	0.00	5.10	45.48
E (Proposed)	✓	✓	✓	0.958	0.037	0.01	0.88	47.21

The symbol “✓” indicates that the corresponding module is included in the experimental configuration, while “×” indicates that the module is excluded.

The CCWL module is a key innovation enabling sub-pixel high-precision localization and false alarm control. Comparing experimental Group D (without CCWL) with the complete algorithm in Group E, it is evident that the CCWL module effectively fuses threshold-exceeding points within the region, reducing the false alarm rate from 95.7% to 3.7% and decreasing the average localization error from 5.10 pixels to 0.88 pixels, while introducing only minimal computational overhead. The results of experimental Group C further confirm that, through connected component extraction, false alarm region suppression, and probability-weighted voting, the CCWL module effectively overcomes the quantization limitations of pixel-level detection, significantly improving localization accuracy while suppressing spurious responses induced by clutter.

The slight P_d reduction from 96.1% (Group D) to 95.8% (Group E) is attributable to a spurious true positive effect. Group D, with its higher P_f , generates false alarms near true targets that appear correct in evaluation. CCWL suppresses these spurious detections, yielding a slightly lower but more reliable P_d , while the dramatic P_f reduction confirms CCWL's effectiveness.

To further clarify the advantage of CCWL, we conducted a quantitative comparison against alternative post-processing localization strategies on the held-out real-world dataset, with the detailed quantitative results presented in Table 10. Max NMS retains only the peak response within each detection window, thereby discarding the surrounding confidence distribution that carries valuable sub-pixel localization information. Density-based spatial clustering of applications with noise (DBSCAN) [34] performs density-based clustering to group adjacent response points, but it requires careful tuning of two critical hyperparameters: the neighborhood radius ϵ and the minimum number of points MinPts. In our experiments, we conducted an exhaustive grid search on the validation set to optimize these parameters, resulting in $\epsilon = 5$ and MinPts = 10. In contrast, CCWL combines connected component analysis with a fixed size threshold derived from the expected soft-label response size and probability-weighted voting. It achieves robust and consistent localization performance across the entire operating SNR range without parameter retuning, and does not rely on assumptions about the specific shape of target response distributions.

Table 10. Performance Comparison of Different Post-Processing Methods on Real-World Dataset.

Post-Processing Method	P_d	P_f	$d_{\min}$	d_{ave}
Max NMS	0.853	0.142	0.00	2.41
DBSCAN	0.960	0.086	0.08	1.84
CCWL	0.958	0.037	0.01	0.88

4.6. Parameter Sensitivity Analysis

The key hyperparameters of F²DN-CCWL are selected through a combination of theoretical derivation and experimental validation. ΔR is set to 32 pixels, which exactly matches the FastDN grid cell size ($288/9 = 32$ pixels), ensuring that the search radius covers the maximum localization uncertainty when detection confidence is low. T_{dis} is set to 5 pixels, which balances the need to merge nearby detections of the same target while preserving distinction between separate targets.

To evaluate the robustness of the proposed method to key hyperparameters and validate our default settings, we conducted a comprehensive sensitivity study on the held-out real-world dataset. Two critical parameters were tested: the detection threshold τ and the connected component size threshold T_{cc} . For each parameter, we varied its value within a reasonable range while fixing the other parameter at its default value. The results are summarized in Table 11.

Table 11. Parameter Sensitivity Analysis Results on Real-World Dataset.

Parameter	Values	P_d	P_f
τ (with T_{cc} fixed at 20)	0.1	0.970	0.526
	0.3	0.963	0.378
	0.5	0.958	0.037
	0.7	0.896	0.018
	0.9	0.688	0.008
T_{cc} (with τ fixed at 0.5)	10	0.961	0.106
	20	0.958	0.037
	30	0.874	0.021

The results demonstrate that both parameters exhibit robust performance across their tested ranges. As expected, the detection threshold τ has a more pronounced impact on the trade-off between detection probability and false alarm rate. Our default settings achieve the optimal overall balance between these two metrics. Furthermore, performance degradation is minimal even when parameters deviate moderately from their default values, confirming the robustness of the proposed method to hyperparameter choices.

These sensitivity results not only validate the robustness of our default parameter settings but also provide valuable insights into the performance characteristics of our method. The consistent performance ranking across different parameter values confirms that the advantages of the proposed F²DN-CCWL framework are inherent to its architectural design, rather than being dependent on specific parameter tuning.

5. Conclusions

To address the performance collapse of traditional CFAR algorithms and the inherent bottleneck of existing deep learning methods in balancing detection accuracy, localization precision, real-time efficiency, and false alarm control for weak target detection in radar Range-Doppler (RD) spectra under strong clutter and low signal-to-noise ratio (SNR) conditions, this paper proposes a progressive sub-pixel intelligent detection algorithm, termed F²DN-CCWL, inspired by the human visual cognitive mechanism of “global scanning-local focusing.” Simulation results demonstrate that across the full SNR range from −20 dB to 0 dB, the proposed algorithm outperforms the conventional CA-CFAR algorithm, as well as the YOLO-based and DCNN-based comparative methods, in terms of both detection performance and localization accuracy. Particularly under the challenging condition of −20 dB SNR, the algorithm achieves a detection probability of 95.3%, a false alarm rate of 3.1%, and an average localization error of 0.76 pixels. Furthermore, real-world experiments using an X-band anti-UAV radar confirm that the algorithm maintains a detection probability of 95.8% and a false alarm rate of 3.7% in complex real environments, exhibiting no significant performance degradation and demonstrating strong robustness. The core source of this performance advantage is that the algorithm does not simply transplant optical image detection networks; instead, it presents a full-chain customized design—encompassing the detection paradigm, network architecture, training strategy, and post-processing algorithm—grounded in the physical signal characteristics of the radar RD spectrum and the underlying mechanisms of weak target detection.

Ablation studies quantitatively demonstrate the complementary roles of the three core modules: FastDN provides high-recall global screening with 12.96 ms latency; FineDN enables precise pixel-level discrimination within local RoIs; and CCWL achieves sub-pixel localization while reducing the false alarm rate from 95.7% to 3.7% with negligible overhead. The integration of these three modules forms a progressive detection pipeline that achieves a multidimensional balance among detection performance, localization accuracy,

and real-time engineering capability, culminating in a single-frame processing latency of 47.21 ms.

We acknowledge that our current real-world validation has certain limitations: it uses only one X-band radar, one UAV target type, and primarily radial round-trip flight trajectories collected in a rooftop scenario. However, our method is designed to leverage universal physical characteristics of RD spectra, and additional simulation results demonstrate consistent performance across different radar parameters and target RCS values. Future work will include validation on more diverse datasets with multiple radar bands, various target types, complex maneuvering trajectories, and severe clutter environments such as urban canyons and coastal regions.

Author Contributions: Conceptualization, M.Q. and J.W.; Methodology, M.Q.; Software, M.Q.; Validation, M.Q., J.W. and G.W.; Formal analysis, M.Q.; Investigation, M.Q.; Resources, J.W. and G.W.; Data curation, M.Q.; Writing—original draft preparation, M.Q.; Writing—review and editing, J.W. and G.W.; Visualization, M.Q.; Supervision, J.W.; Project administration, J.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data that support the findings of this study are available from the corresponding author upon reasonable request, subject to formal institutional review and approval by the Nanjing Research Institute of Electronics Technology. These data cannot be publicly deposited in open repositories because they were collected using a proprietary radar system developed by a state-owned national defense research institute. The system's technical specifications, including range resolution, Doppler resolution, and detection sensitivity, are subject to China's national security and confidentiality regulations for dual-use technologies.

Conflicts of Interest: Author Mingjie Qiu, Jianming Wang and Guangxin Wu were employed by Nanjing Research Institute of Electronics Technology. The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

CCWL	Connected Component Weighted Localization
CFAR	Constant False Alarm Rate
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
DCNN	Deep Convolutional Neural Network
DFS	Depth-First Search
ECA	Efficient Channel Attention
MTI	Moving Target Indication
NMS	Non-Maximum Suppression
PRF	Pulse Repetition Frequency
RD	Range-Doppler
ROC	Receiver Operating Characteristic
RoI	Region of Interest
SNR	Signal-to-Noise Ratio

References

1. Wagner, S.; Johannes, W.; Qosja, D.; Brüggewirth, S. Small Target Detection in a Radar Surveillance System Using Contractive Autoencoders. *IEEE Trans. Aerosp. Electron. Syst.* **2024**, *60*, 51–67. [[CrossRef](#)]
2. Choi, J.; Chun, Y.-H.; Eom, S.C.; Oh, D.; Kim, Y. Enhanced Radar False Alarm Mitigation in Low-RCS Target Detection Using Time-Varying Trajectories on Range-Doppler Diagrams With DCNN. *IEEE Trans. Instrum. Meas.* **2025**, *74*, 2516013. [[CrossRef](#)]

3. Sun, H.; Oh, B.-S.; Guo, X.; Lin, Z. Improving the Doppler Resolution of Ground-Based Surveillance Radar for Drone Detection. *IEEE Trans. Aerosp. Electron. Syst.* **2019**, *55*, 3667–3673. [[CrossRef](#)]
4. Wang, Q.; Wei, X.; Gao, M. LSSDNet: A Low-Slow-Small Target Detection Method Based on Centimeter Wave Radar in the Urban Environment. *IEEE Sens. J.* **2025**, *25*, 22118–22137. [[CrossRef](#)]
5. Jia, F.; Wei, Y.; Yan, L.; Ding, L.; Qian, J. A Light Architecture for Radar Range–Doppler Detection. *IEEE Sens. J.* **2025**, *25*, 36541–36553. [[CrossRef](#)]
6. Hansen, V. Adaptive detection mode with threshold control as a function of spatially sampled clutter-level estimates. *RCA Rev.* **1968**, *29*, 414–464. [[CrossRef](#)]
7. Chen, X.; Liu, K.; Zhang, Z. A PointNet-Based CFAR Detection Method for Radar Target Detection in Sea Clutter. *IEEE Geosci. Remote Sens. Lett.* **2024**, *21*, 3502305. [[CrossRef](#)]
8. Smith, M.E.; Varshney, P.K. Intelligent CFAR processor based on data variability. *IEEE Trans. Aerosp. Electron. Syst.* **2000**, *36*, 837–847. [[CrossRef](#)]
9. Safa, A.; Dalil, B.; Pouzin, F.; Duroy, S. A Low-Complexity Radar Detector Outperforming OS-CFAR for Indoor Drone Obstacle Avoidance. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2021**, *14*, 9162–9175. [[CrossRef](#)]
10. Sahed, M.; Kenane, E.; Khalfa, A.; Djahli, F. Exact Closed-Form Pfa Expressions for CA- and GO-CFAR Detectors in Gamma-Distributed Radar Clutter. *IEEE Trans. Aerosp. Electron. Syst.* **2023**, *59*, 4674–4679. [[CrossRef](#)]
11. Wang, S.; Herschel, R. Fast 3D-CFAR for drone detection with MIMO radars. In Proceedings of the 18th European Radar Conference (EuRAD), London, UK, 28–30 September 2022; pp. 209–212.
12. Ai, J.; Pei, Z.; Yao, B.; Wang, Z.; Xing, M. AIS Data Aided Rayleigh CFAR Ship Detection Algorithm of Multiple-Target Environment in SAR Images. *IEEE Trans. Aerosp. Electron. Syst.* **2022**, *58*, 1266–1282. [[CrossRef](#)]
13. Orlando, D.; Ricci, G. A New CFAR Detector Based on the EM Algorithm. *IEEE Signal Process. Lett.* **2025**, *32*, 2035–2039. [[CrossRef](#)]
14. Zeng, T.; Zhang, X.; Li, M.; Yu, W.; Liu, Q. CFAR-DP-FW: A CFAR-Guided Dual-Polarization Fusion Framework for Large-Scene SAR Ship Detection. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2024**, *17*, 7242–7259. [[CrossRef](#)]
15. Geng, Z.; Yan, H.; Zhang, J.; Zhu, D. Deep-Learning for Radar: A Survey. *IEEE Access* **2021**, *9*, 141800–141818. [[CrossRef](#)]
16. Huang, T.-Y.; Lee, M.-C.; Yang, C.-H.; Lee, T.-S. YOLO-ORE: A Deep Learning-Aided Object Recognition Approach for Radar Systems. *IEEE Trans. Veh. Technol.* **2023**, *72*, 5715–5731. [[CrossRef](#)]
17. Zhai, X.; Huang, Z.; Li, T.; Liu, H.; Wang, S. YOLO-Drone: An optimized YOLOv8 network for tiny UAV object detection. *Electronics* **2023**, *12*, 3664. [[CrossRef](#)]
18. Zhao, W.; Li, X.; Chen, S. BCA-DetNet: Background Contrast-Based Attention Detection Network for UAVs Detection in Pulse-Doppler Radar. *IEEE Trans. Aerosp. Electron. Syst.* **2025**, *61*, 18918–18932. [[CrossRef](#)]
19. He, J.; Wang, W. NST-YOLO: Improved YOLOv10 model for small target UAV detection. *Ain Shams Eng. J.* **2025**, *16*, 103787. [[CrossRef](#)]
20. Wang, C.; Tian, J.; Cao, J.; Wang, X. Deep Learning-Based UAV Detection in Pulse-Doppler Radar. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 5105612.
21. Tian, J.; Wang, C.; Cao, J.; Wang, X. Fully Convolutional Network-Based Fast UAV Detection in Pulse Doppler Radar. *IEEE Trans. Geosci. Remote Sens.* **2024**, *62*, 5103112. [[CrossRef](#)]
22. Wang, J.; Li, S. Maritime Radar Target Detection in Sea Clutter Based on CNN With Dual-Perspective Attention. *IEEE Geosci. Remote Sens. Lett.* **2023**, *20*, 3500405. [[CrossRef](#)]
23. Wang, L.; Tang, J.; Liao, Q. A Study on Radar Target Detection Based on Deep Neural Networks. *IEEE Sens. Lett.* **2019**, *3*, 7000504. [[CrossRef](#)]
24. Li, Y.; Wang, Z.; Chen, H.; Li, Y. A Density Clustering-Based CFAR Algorithm for Ship Detection in SAR Images. *IEEE Geosci. Remote Sens. Lett.* **2024**, *21*, 4009505. [[CrossRef](#)]
25. Gao, J.; Zhang, Y.; Li, Z.; Chen, Y.; Wang, H. An algorithm for road target detection of autonomous vehicles based on improved YOLOv8. *Sci. Rep.* **2025**, *15*, 21061. [[CrossRef](#)] [[PubMed](#)]
26. Safaldin, M.; Zaghdien, N.; Mejdoub, M. An Improved YOLOv8 to Detect Moving Objects. *IEEE Access* **2024**, *12*, 59782–59806. [[CrossRef](#)]
27. Li, Y.; Chen, Z.; Wang, H.; Zhang, L.; Zhao, J. SOD-YOLO: Small-object-detection algorithm based on improved YOLOv8 for UAV images. *Remote Sens.* **2024**, *16*, 3057. [[CrossRef](#)]
28. Zeng, N.; Wu, P.; Wang, Z.; Li, H.; Liu, W.; Liu, X. A Small-Sized Object Detection Oriented Multi-Scale Feature Fusion Approach With Application to Defect Detection. *IEEE Trans. Instrum. Meas.* **2022**, *71*, 3507014.
29. Li, Q.; Xiao, D.; Shi, F. A Decoupled Head and Coordinate Attention Detection Method for Ship Targets in SAR Images. *IEEE Access* **2022**, *10*, 128562–128578. [[CrossRef](#)]
30. Yu, D.; Liu, X.; Chen, Z.; Wu, Y.; Li, J. An Anchor-Free and Angle-Free Detector for Oriented Object Detection Using Bounding Box Projection. *IEEE Trans. Geosci. Remote Sens.* **2023**, *61*, 5618517. [[CrossRef](#)]

31. Li, J.; Zhang, H.; Wang, G.; Chen, J.; Liu, W. Joint Clutter Suppression and Moving Target Indication in 2D Azimuth Rotated Time Domain for Single-Channel Bistatic SAR. *IEEE Trans. Geosci. Remote Sens.* **2023**, *61*, 5202516.
32. Wang, Q.; Wu, B.; Zhu, P.; Li, P.; Zuo, W.; Hu, Q. ECA-Net: Efficient channel attention for deep convolutional neural networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 11531–11539.
33. Li, L.; Chen, Z.; Ma, Y.; Liu, L. A Lightweight High-Resolution Remote Sensing Image Cultivated Land Extraction Method Integrating Transfer Learning and SENet. *IEEE Access* **2024**, *12*, 113694–113704. [[CrossRef](#)]
34. Yoo, T.; Bang, H.; Youn, W. Adaptive DBSCAN-based probabilistic data association filter for maritime object tracking using RADAR. *IEEE Sens. Lett.* **2025**, *9*, 6006404. [[CrossRef](#)]

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.