

Technical Note

# GD-Det: Low-Data Object Detection in Foggy Scenarios for Unmanned Aerial Vehicle Imagery Using Re-Parameterization and Cross-Scale Gather-and-Distribute Mechanisms

Rui Shi <sup>1</sup>, Lili Zhang <sup>2,\*</sup> , Gaoxu Wang <sup>1</sup>, Shutong Jia <sup>3</sup>, Ning Zhang <sup>2</sup> and Chensu Wang <sup>4</sup>

- <sup>1</sup> State Key Laboratory of Hydrology–Water Resources and Hydraulic Engineering, Nanjing Hydraulic Research Institute, Nanjing 210029, China; rshi@nhri.cn (R.S.); gxwang@nhri.cn (G.W.)  
<sup>2</sup> College of Computer Science and Software Engineering, Hohai University, Nanjing 211100, China; 211307040026@hhu.edu.cn  
<sup>3</sup> College of Information Science and Engineering, Hohai University, Changzhou 213212, China; 241613010104@hhu.edu.cn  
<sup>4</sup> College of Information Science and Engineering, Henan University of Technology, Zhengzhou 450001, China; 211011001009@stu.haut.edu.cn  
\* Correspondence: lilzhang@hhu.edu.cn

**Abstract:** Unmanned Aerial Vehicles (UAVs) play an extremely important role in real-time object detection for maritime emergency rescue missions. However, marine accidents often occur in low-visibility weather conditions, resulting in poor image quality and a lack of object detection samples, which significantly reduces detection accuracy. To tackle these issues, we propose GD-Det, a low-data object detection model with high accuracy, specifically designed to handle limited sample sizes and low-quality images. The model is primarily composed of three components: (i) A lightweight re-parameterization feature extraction module which integrates RepVGG blocks into multi-concat blocks to enhance the model's spatial perception and feature diversity during training. Meanwhile, it reduces computational cost in the inference phase through the re-parameterization mechanism. (ii) A cross-scale gather-and-distribute pyramid module, which helps to augment the relationship representation of four-scale features via flexible skip fusion and distribution strategies. (iii) A decoupled prediction module with three branches is to implement classification and regression, enhancing detection accuracy by combining the prediction values from tri-level features. (iv) We also use a domain-adaptive training strategy with knowledge transfer to handle low-data issues. We conducted low-data training and comparison experiments using our constructed dataset AFO-fog. Our model achieved an overall detection accuracy of 84.8%, which is superior to other models.

**Keywords:** low-data; re-parameterization feature extraction; knowledge transfer; cross-scale gather-and-distribute; UAV imagery; object detection



Academic Editor: Salah Bourennane

Received: 23 January 2025

Revised: 13 February 2025

Accepted: 19 February 2025

Published: 24 February 2025

**Citation:** Shi, R.; Zhang, L.; Wang, G.; Jia, S.; Zhang, N.; Wang, C. GD-Det: Low-Data Object Detection in Foggy Scenarios for Unmanned Aerial Vehicle Imagery Using Re-Parameterization and Cross-Scale Gather-and-Distribute Mechanisms. *Remote Sens.* **2025**, *17*, 783. <https://doi.org/10.3390/rs17050783>

**Copyright:** © 2025 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

As maritime industries continue to evolve, marine accidents happen frequently and maritime search and rescue missions have become more crucial [1]. Unmanned Aerial Vehicles (UAVs) have turned into a highly effective and convenient tool for assisting in maritime rescue, because they can reach far or inaccessible areas and have efficient coverage capabilities with low costs [2]. Their ability to operate in challenging environments makes them invaluable for detecting objects in adverse conditions, such as foggy weather. However, detecting maritime objects in foggy environments remains a major challenge due

to low visibility [3], blurred textures [4], and low-labeled datasets [5]. These challenges lead to weak detection performance for deep learning models [6].

Object detection models based on deep learning are widely applied across various fields and are typically categorized into two types: one-stage and two-stage detection models [7,8]. Early two-stage detection models [9–12] achieved high detection accuracy but faced challenges, such as large computational overhead and suboptimal performance, in detecting small objects. To address these issues, models like [13–15] were proposed, utilizing feature fusion techniques to enhance small object detection while balancing detection accuracy and inference speed. Single-stage detection models [16,17] enable end-to-end object localization and classification directly on the input image, offering higher inference speeds compared to two-stage models, though they exhibit some limitations in detection accuracy. Anchor-free models [18–21] mitigate the problem of class imbalance by replacing anchor boxes with key points, thereby improving detection accuracy. Research on applying deep learning to maritime object detection has also been progressively developed. Bousetouane et al. [22] proposed a weak object detection method based on handcrafted features for ship localization and classification. However, marine accidents often occur in low-visibility conditions (e.g., fog), which lead to a slump in image quality and reduce the detection performance of all models [23]. Additionally, the limited availability of sample data for object detection in foggy scenarios further exacerbates the detection challenge.

In response, several solutions have been proposed by researchers. Ma et al. [24] proposed the YOLOX model, which improves object detection performance under adverse weather conditions through joint image restoration and domain adaptation frameworks. Zhang et al. [25] introduced the SG-Det model, which enhances detection speed and accuracy for UAV-based maritime search and rescue tasks by incorporating a channel shuffle operation and a lightweight object detector. To address the issue of data scarcity, Yue et al. [26] combined MobileNetv2, YOLOv4, and sparse training techniques to propose a lightweight ship detection method aimed at solving the problem of low data availability. However, they all face the challenge of not being able to simultaneously address both low data availability and high image quality. To further improve model performance, many challenges remain to be overcome.

Therefore, to tackle the above challenges like low-data and low image quality in low-visibility weather conditions, this paper introduces a low-data object detection model GD-Det in foggy scenarios for UAV imagery based on re-parameterization and cross-scale gather-and-distribute mechanisms. The primary contributions of this paper are as follows:

- (1) We introduce a re-parameterization feature extraction module integrating RepVGG blocks into multi-concat blocks, which can improve the model's spatial perception and feature diversity during training while reducing parameter costs during inference.
- (2) We propose a cross-scale gather-and-distribute pyramid module with a self-attention mechanism and injection algorithm which can augment the relationship representations of four-scale features via flexible skip fusion and distribution and maximize the utilization of global and local features.
- (3) We conduct a decoupled prediction module with three branches which implements classification and regression, respectively, and enhances the detection accuracy via the fusion of the prediction accuracy of tri-level features.
- (4) We use a domain-adaptive training strategy with knowledge transfer to cope with low-data issues which uses the knowledge transfer and feature alignment in cross-domain scenarios and less samples to make the model convergence.

## 2. Related Work

### 2.1. Lightweight Networks

The convolutional neural networks (CNNs) usually need plenty of samples to train and reach convergence, which not handle the low-data detection issues, so the lightweight networks [27] have been introduced to minimize the model's parameters and computational complexity while ensuring high accuracy using a few samples. Classical lightweight networks include ShuffleNet [28], ESPNet [29], and MobileNet [30]. ShuffleNetV1 [28] optimizes the information flow in grouped convolutions by introducing the channel shuffle operation, thereby accelerating the network performance. ShuffleNetV2 [31] acts as a channel split operation to modify and improve ShuffleNetV1. ESPNet improves semantic segmentation [32] accuracy significantly while maintaining low computational complexity by introducing the Efficient Spatial Pyramid module. MobileNets [30] involve a class of lightweight deep neural networks designed for efficient mobile and embedded vision applications. The core idea behind MobileNets is the use of depthwise separable convolutions. This technique decomposes the standard convolution operation into two smaller operations: depthwise convolution and pointwise convolution. This significantly reduces the number of parameters and computations compared to traditional convolutions, making the network more efficient. MobileNetV2 [33] introduces a Linear Bottleneck and Inverted Residual structure to optimize the network. Compared to MobileNetV2, MobileNetV3 [34] further optimizes the network by removing convolutional layers such as  $3 \times 3$  DConv and  $1 \times 1$  convolutions.

In recent years, new and improved lightweight networks have emerged. SMFANet [35] is a lightweight self-modulation feature aggregation network designed for efficient image super-resolution. It addresses the challenges of balancing high reconstruction performance with computational efficiency, particularly in resource-constrained environments. XFeat [36] proposes a lightweight CNN architecture capable of keypoint detection and local feature extraction while maintaining computational efficiency. It uses innovative network layer distribution and lightweight convolutional blocks to reduce computational burden. These modifications reduce computational complexity and parameters without compromising model accuracy.

Because the lightweight networks have the advantage of maintaining detection accuracy and inference speed with low-data, they have been increasingly employed for object detection in foggy weather issues as a result. Zhang et al. [37] proposed XM-YOLOViT, which integrates transformer modules into a lightweight YOLO framework to improve detection robustness in foggy weather. By leveraging transformer-based attention mechanisms, XM-YOLOViT enhances feature extraction, achieving superior detection accuracy for fog-obscured objects, especially for small and partially occluded objects, while maintaining real-time inference. Hou et al. [38] optimized YOLOv5 architecture by introducing the RFCBAM module, which focuses on spatial features within the receptive field, thereby improving the model's ability to extract effective features under foggy conditions. Zhang et al. [39] optimized the network structure by improving its interpretability and reducing the number of network layers, proposing a lightweight pixel-wise semantic segmentation model that achieves high-precision and high-speed waterbody extraction from GF-1 remote sensing images. Our proposed lightweight model ensures spatial awareness and feature diversity while reducing computational overhead during inference, enhancing detection accuracy and demonstrating superior performance under foggy conditions.

### 2.2. Re-Parameterization and Multi-Scale Feature Fusion

Structural re-parameterization models decouple the network architecture for training and inference with different backbones. They use a multi-branch network in training

to extract more features and a single-branch structure in inference to reduce parameters and computational costs. The RepVGG model [40] uses structural re-parameterization to separate the training and inference stages, optimizing the network for both accuracy and efficiency. By combining the advantages of single-branch and multi-branch architectures, RepVGG achieves fast inference speeds while maintaining high detection accuracy.

Ding et al. [41] also introduced a universal convolutional module, the Diverse Branch Block (DBB), based on structural re-parameterization, which also enhanced model performance while maintaining inference efficiency. Kang et al. [42] proposed a novel object detection architecture called RCS-YOLO, and the re-parameterization in the RCS-YOLO model enables the transformation of complex multi-branch learning channel Shuffles into a simplified single-branch inference structure. Zhang et al. [43] introduced re-parameterization to a transformer structure to address performance degradation and enhance the effectiveness of transformer models. Hence, a re-parameterization-based network is one better new strategy for balancing the detection accuracy and low reference cost, and it always demonstrates exceptional performance combined with different network architectures.

Feature Pyramid Networks (FPNs) [44] were designed to solve the feature fusion issue with different scales, including high computational costs, insufficient semantic information, and inconsistencies between training and inference. By constructing multi-scale feature pyramids, an FPN effectively overcomes these issues and significantly improves the recall and detection accuracy for small objects. However, FPN-like structures are unable to transmit information without loss during the information fusion process, which is one of the main reasons to hinder the YOLO series from achieving better performance. Wang et al. [45] proposed an object detection model Gold-YOLO with self-attention operations, which avoided the inherent information loss of traditional FPNs and improved information fusion in the network's neck. However, Gold-YOLO's detection accuracy is less effective in foggy scenarios. Building on this, we will develop a cross-scale skip fusion and gather-and-distribute pyramid module among different scales to augment the model's modeling ability in regard to relationships. Furthermore, we will seamlessly integrate RepVGG blocks into multi-concatenation blocks, creating a multi-branch architecture in the training phase to facilitate comprehensive learning. In the reference phase, we will adopt a single-branch structure to balance accuracy and computational cost under foggy conditions.

### 3. Methodology

#### 3.1. Overall Framework

The architecture of the GD-Det model is shown in Figure 1. This model adopts a one-stage detection framework comprising three primary components: the backbone, the neck, and the head. The backbone incorporates a lightweight re-parameterized feature extraction module which integrates RepVGG blocks within multi-concatenation blocks. This design enhances the model's spatial perception and feature diversity during the training phase while minimizing parameters and computational costs during inference. The module extracts features at four different scales, improving spatial awareness and adaptability to objects of varying sizes. In the neck part, a cross-scale gather-and-distribute pyramid module is designed, employing flexible skip fusion and distribution operations to uniformly collect, fuse, and distribute features, thereby improving the model's ability to capture relationships. Additionally, a self-attention mechanism is incorporated to strengthen the connections between global and local features, enhancing feature representation and information fusion. Finally, in the detection head part, a decoupled prediction module with three branches is constructed, separating classification and regression tasks while combining predictions from tri-level features.

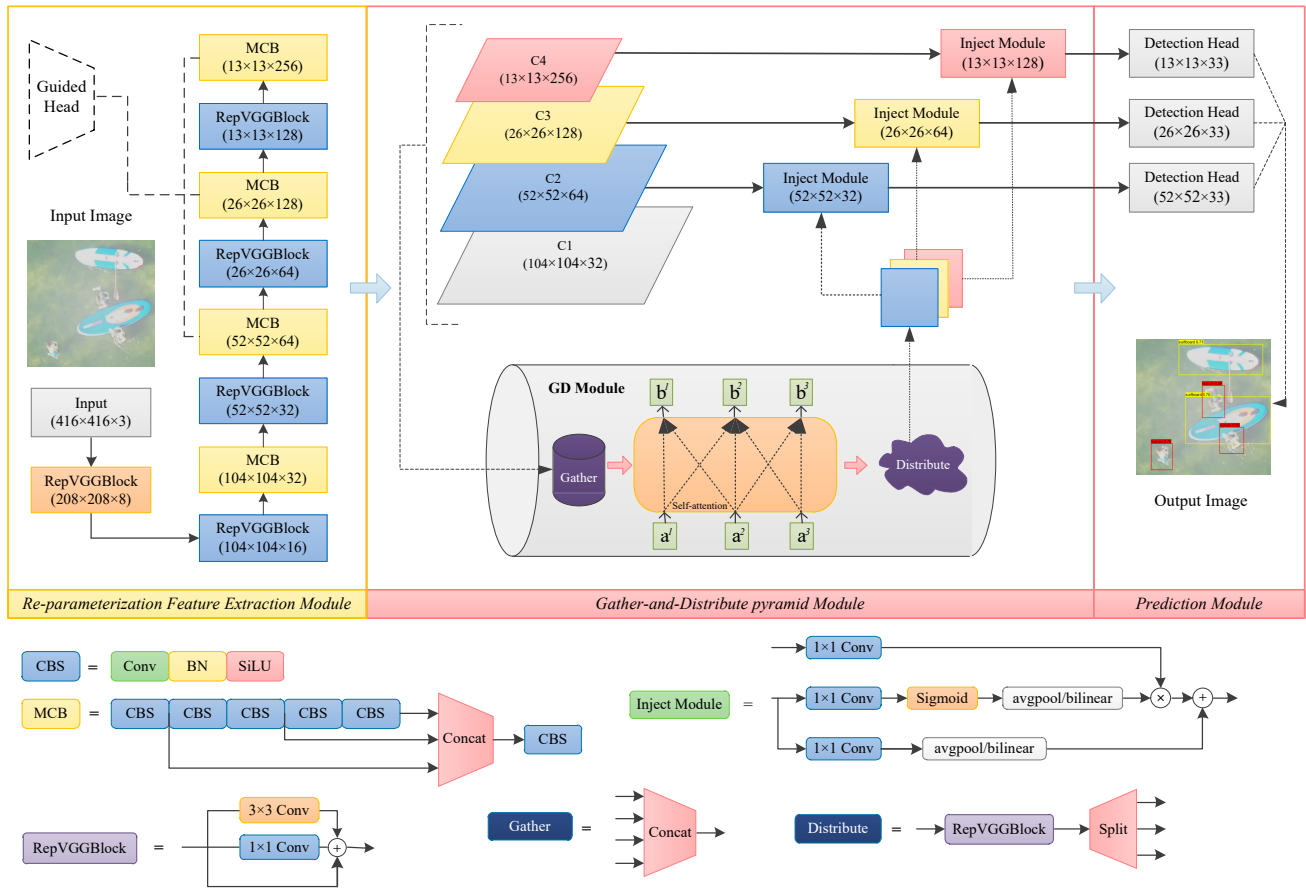


**Figure 1.** The network structure of a GD-Det model.

### 3.2. Lightweight Re-Parameterized Feature Extraction Module

The single-branch network structure offers benefits such as fast inference speed and low storage requirements, but it is often constrained by challenges like gradient explosions. In contrast, the multi-branch network structure demonstrates high versatility and detection accuracy, but it suffers from low parallelism and significant computational and memory demands. To tackle these issues, we introduce a re-parameterized feature extraction module based on the RepVGG network. This module is built by stacking multiple RepVGG blocks and Multi-Concat-Blocks (MCBs), with the MCBs serving as a key component of the backbone network, designed to strengthen multi-level feature extraction capabilities. During the training phase, we utilize a multi-branch structure composed of  $3 \times 3$  convolutions,  $1 \times 1$  convolutions, and residual connections, with batch normalization (BN) layers applied after each branch to improve stability through regularization. This multi-branch design enhances the model's ability to learn complex features by providing diverse gradient paths and more feature representations. In the inference phase, these branches are re-parameterized to a single  $3 \times 3$  convolutional matrix. This process simplifies the computation and, at the same time, preserves the robustness acquired during the training process. This structure conversion not only reduces computational overhead but also maintains high detection accuracy, effectively balancing performance and efficiency. The multi-branch structure allows the model to capture a wider range of feature variations, thereby improving its ability to generalize across diverse datasets. The residual connections further mitigate gradient-related issues, ensuring stable and efficient training. By converting the multi-branch structure into a single-branch structure during inference, the model achieves significant improvements in inference speed and memory efficiency.

without sacrificing accuracy. This is particularly advantageous for real-time applications and deployment on resource-constrained devices.

The MCB module is designed to capture more semantic information from objects, textures, and spatial relationships in the images. It improves the model's ability to understand and represent complex patterns, thereby enhancing the feature-learning capability of the RepVGG block, especially for limited training samples. As shown in Figure 2, the MCB module consists of multiple Conv\_BN\_SiLU (CBS) blocks arranged in both parallel and serial configurations. It features three paths that extract deeper features from lower-level ones at different levels of information granularity. This hierarchical feature extraction mechanism ensures that the model can effectively leverage multi-scale information, further boosting its performance.

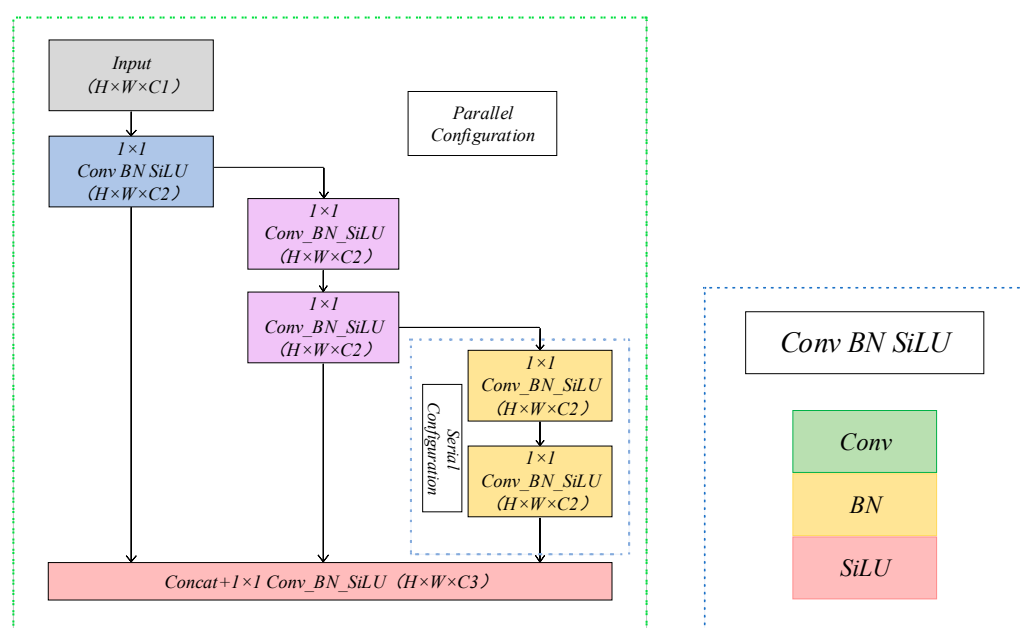


**Figure 2.** Schematic diagram of the multi-concat block module.

### 3.3. Cross-Scale Gather-and-Distribute Pyramid Module

When the training data are limited, the lightweight object detection models usually struggle with insufficient feature extraction capabilities, resulting in more false detections or miss detections. Moreover, in adverse weather conditions, image features like edges and textures can become blurred, leading to a further reduction in detection accuracy. The primary challenge in such issues is to effectively retain and extract the limited available feature information. To tackle these issues, we present a gather-and-distribute pyramid structure that integrates a gather-and-distribute mechanism with a self-attention module. The gather-and-distribute mechanism first unifies and fuses multi-scale features before redistributing them across different scales, while the self-attention module captures the dependency relationships between global and local features. The self-attention module ensures that local features, especially those associated with small objects, are preserved even after multi-scale fusion by dynamically adjusting attention weights to emphasize relevant local details.

Generally, pyramid structures are used in the neck network of object detection models for the fusion of multi-scale features. These structures are composed of multiple branches to integrate features from different scales. However, they can only directly fuse features from adjacent layers, relying on recursive operations to indirectly integrate information from non-adjacent ones. Consequently, the features from each layer can only support adjacent

layers, weakening their ability to enhance features from non-adjacent layers and ultimately reducing the overall effectiveness of feature integration.

In order to effectively utilize the relationships among cross-scale features, this paper proposes a novel gather-and-distribute pyramid module. This module leverages a sophisticated gather-and-distribute mechanism to seamlessly fuse and unify features across different scales. The fused features are then meticulously redistributed to various hierarchical levels, ensuring thorough and comprehensive feature integration via the entire scale spectrum. The gather-and-distribute pyramid module comprises three distinct branches, each designed to handle features with differing levels of granularity. Within each branch, a gather-and-distribute module (GD Module) is used to efficiently collect and merge features, while an inject module follows to facilitate the precise redistribution of these fused features. These components are connected in a serial manner, enabling a streamlined and effective process for feature collection, fusion, and redistribution.

Figure 3 depicts the three serial components of the GD Module: gathering, self-attention, and distribution. In the gathering stage, input feature maps from four different scales are aligned to a uniform size using bilinear interpolation and average pooling operations. Subsequently, these maps are concatenated to form comprehensive multi-scale features. The self-attention part dynamically assigns weights based on the intrinsic dependencies within these multi-scale concatenated features via a self-attention mechanism [46], learns these dependencies and computes attention weights to effectively highlight their importance and correlation, and captures the significant relationships among features at different scales as a result. In the final distribution stage, the extracted features are split into three groups along the channel dimension and injected into different layers. This design obviates feature exchange and fusion between adjacent layers.

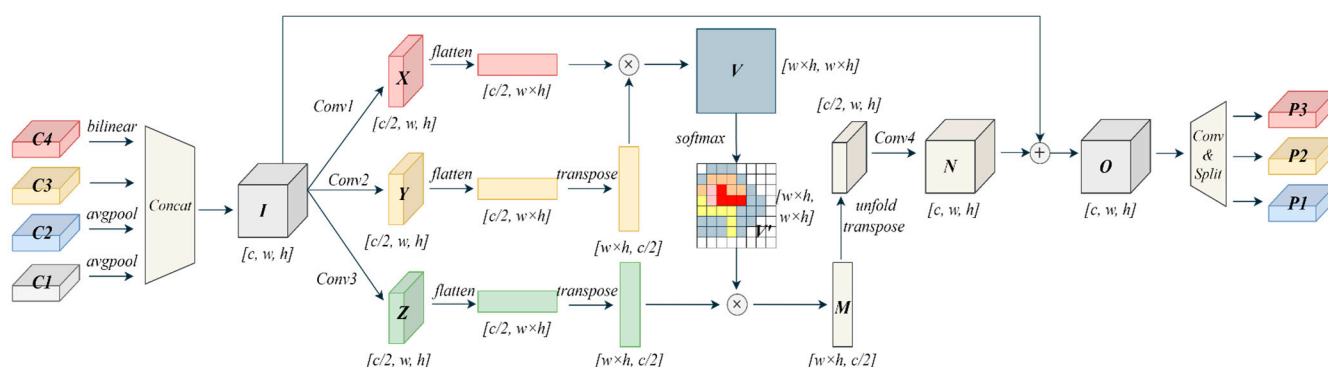


**Figure 3.** Schematic diagram of the gather-and-distribute module. Different colors in the grid represent different weight values, and the darker the color, the greater the weight.

Therefore, the GD Module efficiently integrates and refines multi-scale features, enabling the model to focus on the most relevant characteristics within each scale's features. This module boosts both feature representation and the model's generalization performance in complex scenarios.

The inject module uses a self-attention mechanism to inject global information across various layers, enabling the network to effectively utilize global context and enhance its capability to process and understand local information. Within this module,  $x_{local}$  represents the local information of the current layer, whereas  $y_{global}$  denotes the global information generated by the gather-and-distribute module. Due to the size disparity between local and global information, scaling operations are applied to ensure effective fusion. Specifically, bilinear interpolation, average pooling, and  $1 \times 1$  convolution are employed to scale the information, aligning their feature dimensions. Subsequently, a self-attention mechanism is applied through matrix multiplication between the local and global

features, directing the processing of local features according to the global context. Finally, a residual connection is utilized to propagate global information, ensuring information integrity and stability. As illustrated in Figure 4, this design further improves the model's representational ability and generalization capacity.

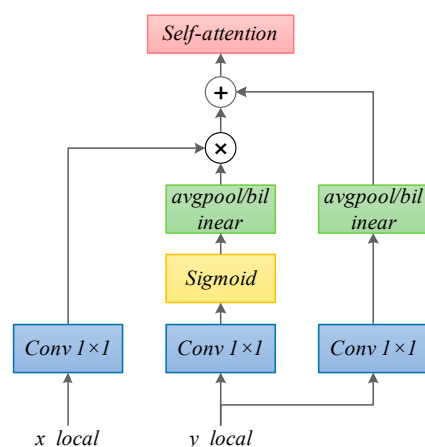


**Figure 4.** Diagram of the inject module.

### 3.4. Decoupled Prediction Module with Three Branches

In conventional object detection models, localization and classification tasks are typically treated as a single unified task. However, a conflict arises between these two goals, that is, an accurate object localization task requires the network to concentrate on fine-grained details and positional information, while a classification task demands that attention is paid to the global context surrounding the target. In order to address this issue, inspired by multi-task learning, we decouple localization and classification into separate branches, as shown in Figure 5. For the localization branch, convolutional layers are employed to preserve spatial details and capture local critical features for precise localization. In the classification branch, we utilize fully connected layers to learn abstract global features through parameter sharing, enabling the network to understand context relationships. Finally, the object branch achieves more robustness and accurate predictions.

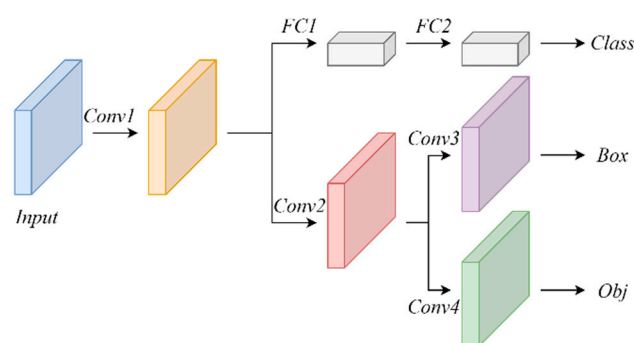


**Figure 5.** Schematic diagram of decoupled prediction module.

### 3.5. Domain-Adaptive Training Strategy with Knowledge Transfer and Feature Alignment

Typically, we use transfer learning to handle the issue that the source domain has ample training samples while the target domain has fewer. However, inappropriate transfer can undermine the model's robustness and stability. Hence, a domain adaptation transfer strategy is applied to allow the model to quickly adapt to the distribution and attributes of the target domain data while maintaining its generalization capability.

In our paper, we use the dataset AFO (Aerial dataset of Floating Objects, AFO) as the source domain and the mini foggy weather dataset mini-AFO-fog we made based on AFO

as the target domain. We also design a domain-adaptive training strategy with knowledge transfer which leverages the general knowledge learned from the source domain during training, reducing reliance on and the importance of the source dataset. This is depicted in Figure 6, with the detailed training process outlined below:

- (1) Pre-training on the source dataset and general knowledge extraction: The backbone of the low-data object detection model GD-Det with three branches is initially trained on the large-scale dataset AFO using a guide head on knowledge extraction to mine the general features of the source domain. The resulting weight parameters are then saved as pre-trained weights.
- (2) Joint training on the target domain dataset: The full GD-Det network, including the detection head and guide head, is then jointly trained on the dataset mini-AFO-fog. This step adapts and aligns the general features from the source domain with those of the target domain, resulting in the global weights of the network.
- (3) Optimization for inference phase: To minimize computational overhead during the inference phase, the guide head used in training is removed during the inference phase, minimizing the quantity of reference parameters and preserving high detection accuracy.

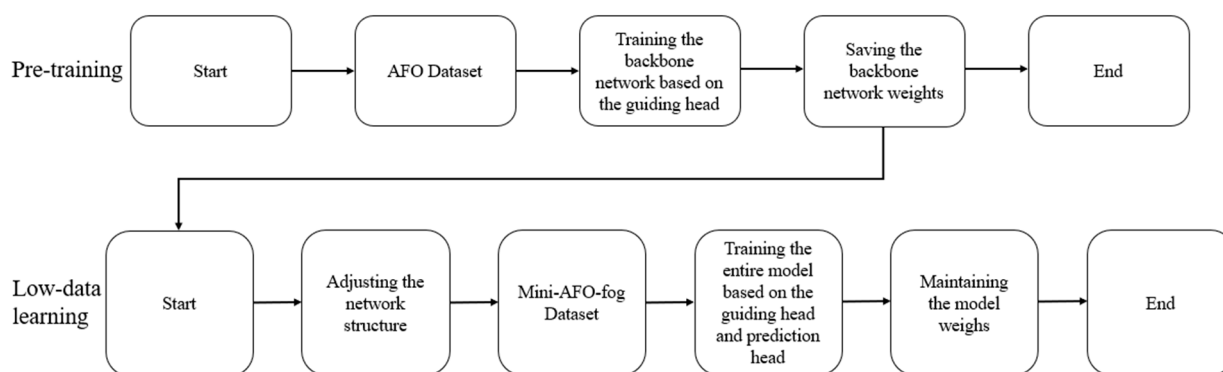


**Figure 6.** Domain-adaptive training strategy.

In the joint training phase, the underlying feature representations are shared across different tasks, allowing the model to learn more generalized and abstract feature expressions. The total loss function for joint training can be expressed as

$$L_{total} = L_{cls}^s + L_{loc}^s + \lambda \left( L_{cls}^t + L_{loc}^t + L_{align} \right) \quad (1)$$

where  $L_{cls}^s$  and  $L_{loc}^s$  represent the classification and localization losses in the source domain, respectively, while  $L_{cls}^t$  and  $L_{loc}^t$  represent the classification and localization losses in the target domain.  $L_{align}$  denotes the feature alignment loss to ensure that features from the source and target domains are aligned in the feature space. The parameter  $\lambda$  is a hyperparameter used to balance the losses between the source and target domains.

The discrepancies in feature distribution between the source and target domains arise from variations in environmental conditions and image quality. To address this issue, feature alignment is applied to minimize the disparity, ensuring that the feature distributions of both domains are as similar as possible. This technique enhances the model's ability to generalize effectively in the target domain. The loss function used for feature alignment is expressed as

$$L_{align} = L_{MMD} = \left\| \frac{1}{n} \sum_{i=1}^n \phi(x_i) - \frac{1}{m} \sum_{j=1}^m \phi(y_j) \right\| \quad (2)$$

where  $\phi(\cdot)$  denotes the mapping function that transforms the samples into the feature space, while  $x_i$  and  $y_j$  represent the samples from the source and target domains, respectively.  $n$  and  $m$  indicate the number of samples in the source and target domains, respectively.

## 4. Experiments and Analysis

### 4.1. Model Training

The hardware configuration for this experiment is as follows: the system is equipped with an AMD R7-5800H CPU, which has a base clock frequency of 3.2 GHz, 16 GB of RAM, and an NVIDIA RTX 3060 GPU with 6 GB of VRAM. The operating system is 64-bit Windows 11, and the deep learning framework used is PyTorch 1.8.2, with CUDA 11.1 facilitating parallel computing. The training parameters are summarized in Table 1. To accommodate hardware limitations and optimize model performance, the input image size is set to  $416 \times 416$ . The batch size for each training iteration is 16, and the model is trained for a total of 200 epochs. The initial learning rate is set to 0.01, with a cosine learning rate decay strategy to adjust the learning rate dynamically. The optimizer used is stochastic gradient descent (SGD), with a weight decay of 0.0005 and a momentum of 0.937, aimed at minimizing the loss function.

**Table 1.** Parameter settings.

| Parameter    | Specific Information |
|--------------|----------------------|
| Input Size   | $416 \times 416$     |
| batch-size   | 16                   |
| epoch        | 200                  |
| lr           | 0.01                 |
| lr decay     | cos                  |
| optimizer    | sgd                  |
| weight decay | 0.0005               |
| momentum     | 0.937                |

We train the GD-Det model on the small-sample foggy dataset min-AFO-fog, and the corresponding loss function curve throughout the training process is illustrated in Figure 7.

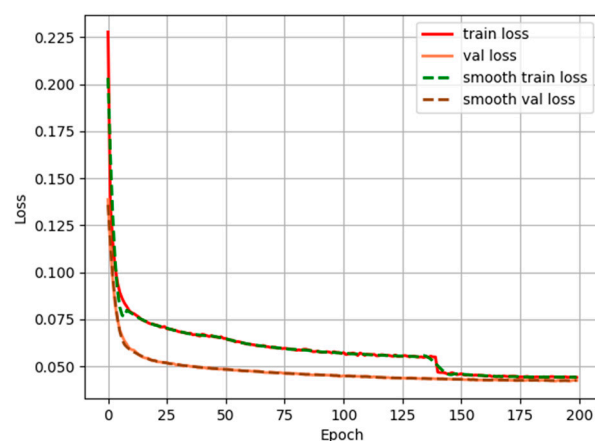


**Figure 7.** The loss function curve of the GD-Det model.

### 4.2. Dataset

#### (1) AFO

The source domain dataset AFO used in our experiments was created by Gasienica-Jozkow et al. [47] using fifty video clips. These videos were captured by various drone-mounted cameras (from  $1280 \times 720$  to  $3840 \times 2160$  resolutions). The videos contained

objects floating on the water surface. A total of 3647 images were extracted and manually annotated, comprising 39,991 objects. The dataset consists of six object categories: human, surfboard, sailboat, buoy, boat, and kayak, as shown in Figure 8. These have been then split into three parts: the training (67.4% of objects), the test (19.12% of objects), and the validation set (13.48% of objects).

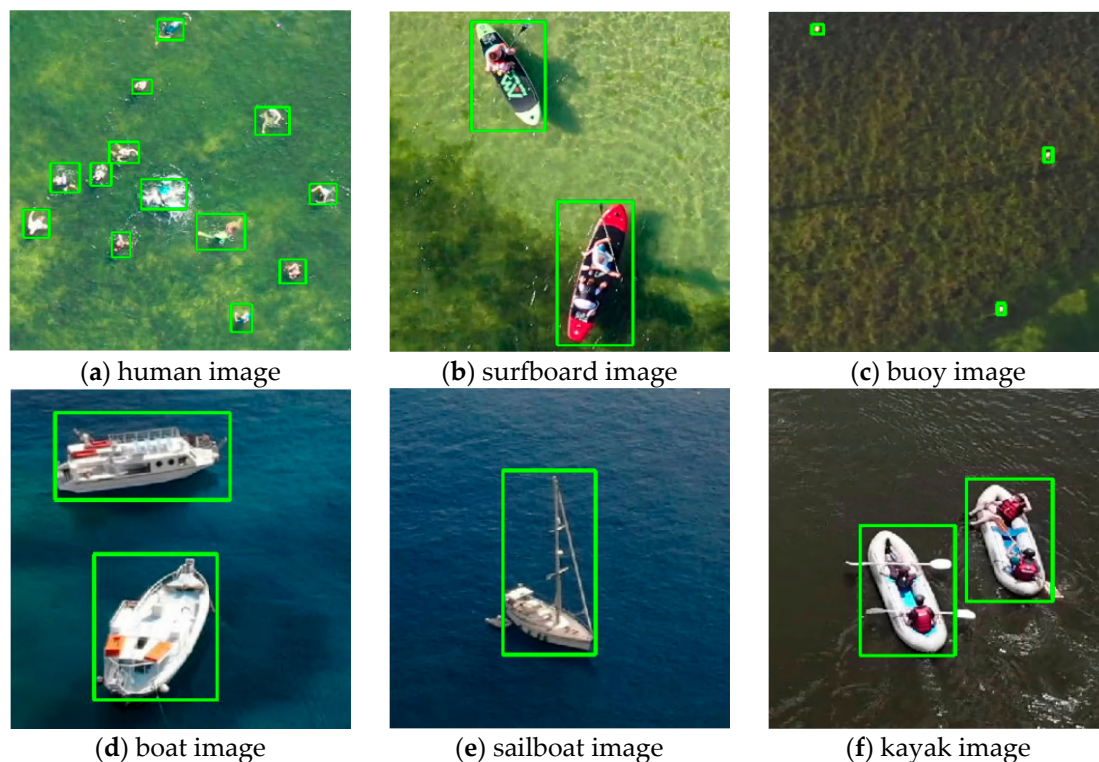


**Figure 8.** The dataset of AFO.

## (2) mini-AFO-fog

Acquiring maritime drone image datasets under adverse weather conditions is often challenging, resulting in a limited availability of samples. To address this limitation, a channel-based fog synthesis algorithm [48] was employed to generate foggy images through adjustments to the image's color channels. This method replicates the physical characteristics of fog while maintaining image realism. The quality of the synthesized foggy images was assessed using a Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity (SSIM), with a PSNR threshold of 30 dB being used to distinguish between light and dense fog conditions. In this paper, the images from the original AFO dataset are divided into three parts: one-third are processed with dense fog effects, one-third with light fog effects, and one-third are left untreated, as illustrated in Figure 9. This approach enables the model to learn to adapt to different weather conditions, thereby enhancing its robustness and generalization ability.

To meet the research requirements, the dataset was annotated using XML files and filtered, yielding a total of 4411 images with a resolution of  $416 \times 416$  pixels. The dataset, after applying the channel synthesis fog algorithm and small-sample data processing, was named mini-AFO-fog. The specific sample sizes of the constructed dataset are shown in Table 2. These images were then split into training, validation, and test sets, with an 8:1:1 ratio for model training and evaluation. It should be specifically noted that, in order to verify that the model parameters are not obtained through overfitting, the validation data were randomly divided into three equal parts for experimentation.



**Figure 9.** Synthetic images in foggy weathers.

**Table 2.** Information of the mini-AFO-fog dataset.

| Object Categories | Sample Size |
|-------------------|-------------|
| Human             | 1249        |
| Surfboard         | 631         |
| Boat              | 566         |
| Buoy              | 530         |
| Sailboat          | 507         |
| Kayak             | 628         |

#### 4.3. Comparison Experiments

To validate the effectiveness of the proposed GD-Det object detection model, we compared it against several conventional lightweight object detection models, including YOLOv5-s, YOLOv7-tiny, YOLOX-tiny, YOLOv8-s, EfficientDet, NanoDet, and RetinaNet, under identical experimental settings and parameter configurations. Multiple evaluation metrics were employed to ensure an accurate and objective assessment of model performance. Specifically, we employed mean Average Precision (mAP) to assess detection accuracy. Mean Average Precision at IoU 0.5 (mAP@0.5) calculates the mean Average Precision when the IoU threshold is set to 0.5. It is generally considered the most commonly used evaluation metric in object detection. Mean Average Precision at IoU 0.9 (mAP@0.9) is used to evaluate the model's ability to make precise, high-quality detections. Frames Per Second (FPS) is used to measure inference speed, and GFLOPs, along with parameter count (Param) are used to evaluate computational complexity and model size. Precision denotes the proportion of correct samples in the predicted results. Recall denotes the proportion of positive samples that are correctly detected in the predicted results. A summary of the experimental results is provided in Table 3.

Compared to YOLOv5-s, YOLOv7-tiny, EfficientDet, and RetinaNet, GD-Det demonstrates advantages on multiple metrics, including detection accuracy, parameter count, computational cost, and inference speed. When compared to YOLOX-tiny, GD-Det achieves a 3.9-point improvement in mAP, albeit with a slight reduction in inference speed, resulting in a 1.5 FPS decrease. Against YOLOv8-s, GD-Det trails by 0.9 points in accuracy but offers significantly lower computational and storage requirements, with YOLOv8-s incurring over 3.1 times the overhead. Relative to the ultra-lightweight NanoDet model, GD-Det outperforms it by 7.2 points in accuracy, and our method can achieve real-time detection as

well. The precision and the mAP of the GD-Det model reaches 84.8%, which indicates our system can detect floating debris in foggy weather conditions at sea.

**Table 3.** Performance comparison of seven methods. The bold sections represent the best data in this group.

| Model        | mAP          | mAP@0.5      | mAP@0.9      | FPS         | GFLOPs       | Param        | Precision   | Recall      |
|--------------|--------------|--------------|--------------|-------------|--------------|--------------|-------------|-------------|
| YOLOv5-s     | 79.7%        | 85.0%        | 74.4%        | 27.2        | 6.7 G        | 7.0 M        | 85.3        | 88.1        |
| YOLOv7-tiny  | 83.2%        | 88.0%        | 80.5%        | 35.2        | 5.5 G        | 6.0 M        | 80.9        | 78.1        |
| YOLOX-tiny   | 80.9%        | 85.5%        | 75.7%        | 38.0        | 6.4 G        | 5.0 M        | 63.4        | 59.8        |
| YOLOv8-s     | 84.7%        | 89.0%        | 80.6%        | 33.4        | 11.6 G       | 11.1 M       | 87.1        | 92.3        |
| EfficientDet | 83.0%        | 87.5%        | 78.1%        | 19.2        | 4.6 G        | 3.8 M        | 90.2        | 89.8        |
| NanoDet      | 77.6%        | 82.5%        | 70.4%        | <b>42.1</b> | <b>2.4 G</b> | <b>2.0 M</b> | 83.0        | 81.9        |
| RetinaNet    | 83.8%        | 87.6%        | 79.4%        | 37.8        | 4.8 G        | 15 M         | 80.1        | 79.5        |
| GD-Det       | <b>84.8%</b> | <b>89.2%</b> | <b>82.9%</b> | 36.5        | 3.7 G        | 3.2 M        | <b>92.2</b> | <b>94.1</b> |

In conclusion, while GD-Det exhibits slightly lower performance in certain metrics compared to specific models, it achieves a well-balanced trade-off across multiple performance metrics. GD-Det significantly reduces both storage and computational overhead while maintaining high detection accuracy and inference speed, making it a robust and efficient solution for low-data object detection tasks.

To visually understand the detection results of various lightweight object detection models, Figure 10 presents the results of the comparative experiments. It is shown that other object detection models have more missed detections or false detections, and the proposed GD-Det model accurately identifies the human contours at the bottom of the image. This highlights GD-Det's superior performance and reliability for low-data object detection tasks in foggy weather or low-visibility weathers in application.

#### 4.4. Ablation Experiments

To validate the contribution of each module in the proposed method, ablation experiments were conducted on the dataset mini-AFO-fog with identical experimental settings and configurations. Modules were incrementally integrated into the model to evaluate their individual contributions. The results of the ablation experiments are shown in Table 4.

Model 1 represents the baseline model RepVGG without any improvement strategies, where the fully connected layer is replaced with a detection head for object detection tasks. Model 2 builds upon Model 1 by introducing the multi-concat block (MCB), which employs multiple branches to extract features at different scales and levels. This enhancement results in 4.8% improvement in detection accuracy. Model 3 further enhances Model 2 by incorporating the gather–distribution pyramid module for efficient and accurate feature aggregation and distribution, as well as a self-attention mechanism to capture dependencies among the input sequence. These improvements lead to a 15% increase in detection accuracy compared to RepVGG.

Model 4 refines Model 3 by decoupling the conventional detection head, enabling the network to independently handle object localization and classification tasks. This modification enhances flexibility and generalizability, leading to a 2.3% increase in detection accuracy and a 4.4 FPS boost in inference speed. Finally, the proposed GD-Det model builds upon Model 4 by introducing a guiding head during training and employing a domain-adaptive training strategy to transfer knowledge from the source domain to the target domain. The detection accuracy reaches 85%, with an inference speed of 36FPS, demonstrating robust recognition capabilities in low-visibility weather conditions. This shows that our proposed GD-Det is an efficient method for detecting maritime objects.

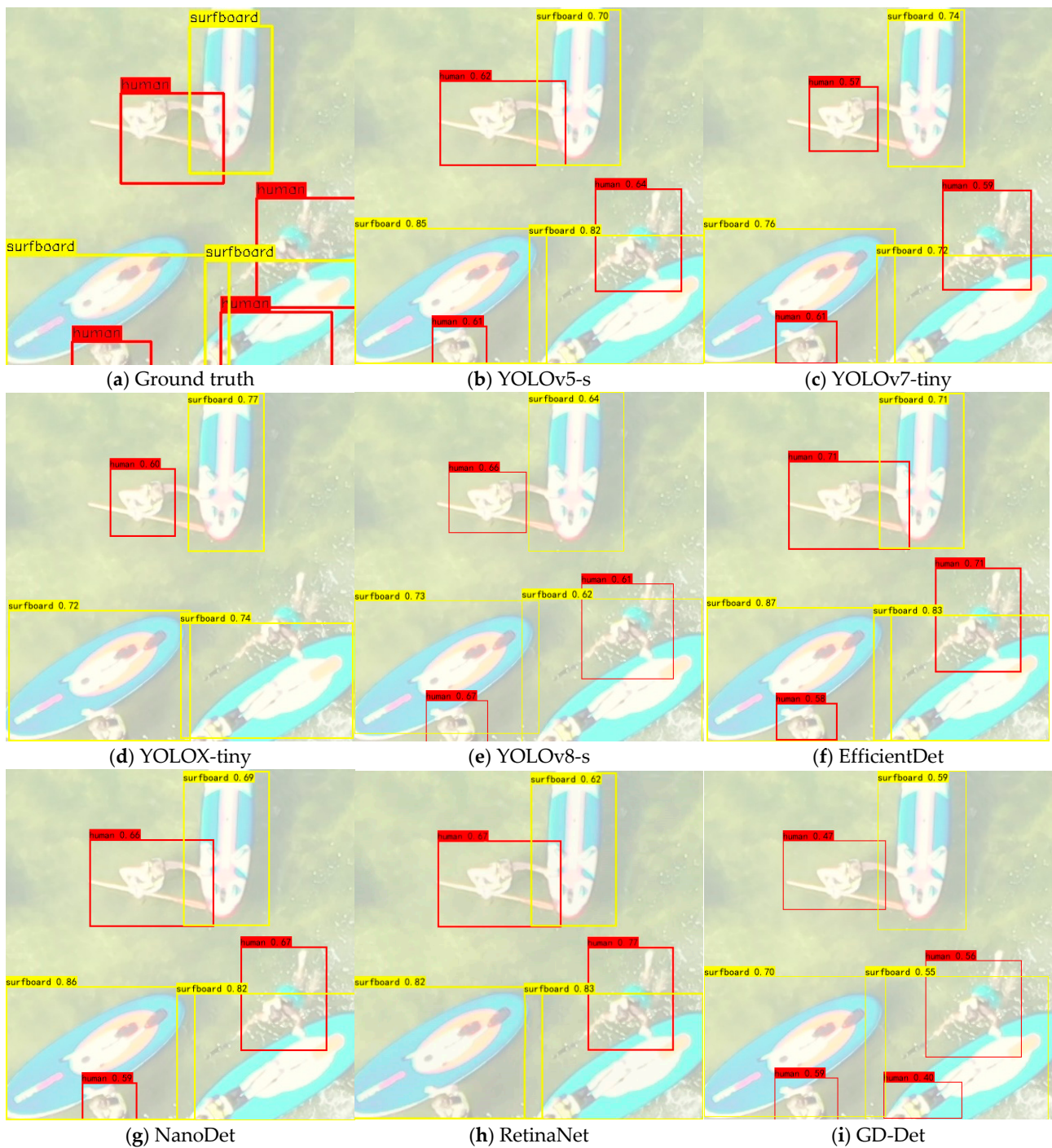


**Figure 10.** The comparison of different lightweight networks. The red boxes indicate that the identified object is “human”, while the yellow boxes indicate that the identified object is “surfboard”.

**Table 4.** Ablation experiments. The “✓” indicates that the model is tested under these conditions. The bold sections represent the best data in this group.

| Model   | RepVGG | MCB | Gather Distribute | Decoupled Head | Guided Head | mAP          | FPS         | GFLOPs       | Param        |
|---------|--------|-----|-------------------|----------------|-------------|--------------|-------------|--------------|--------------|
| Model 1 | ✓      |     |                   |                |             | 65.0%        | 27.3        | <b>1.1 G</b> | <b>0.8 M</b> |
| Model 2 | ✓      | ✓   |                   |                |             | 70.2%        | 31.9        | 1.3 G        | 1.0 M        |
| Model 3 | ✓      | ✓   | ✓                 |                |             | 80.2%        | 31.8        | 3.3 G        | 2.9 M        |
| Model 4 | ✓      | ✓   | ✓                 | ✓              |             | 82.5%        | 36.2        | 3.7 G        | 3.2 M        |
| GD-Det  | ✓      | ✓   | ✓                 | ✓              | ✓           | <b>84.8%</b> | <b>36.5</b> | 3.7 G        | 3.2 M        |

To better illustrate the detection results of each module, Figure 11 shows the visualization outcomes of the ablation experiments. In both Model 1 and Model 2, significant instances of missed detections for human objects are observed. In Model 3, the feature extraction and enhancement achieved through the distribution mechanism and self-attention module effectively mitigate these missed detections. Model 4 further decouples the classification and regression tasks, enhancing detection accuracy while maintaining precise target localization. Finally, the proposed GD-Det achieves superior recognition performance, demonstrating its strong capabilities in real-world detection tasks.

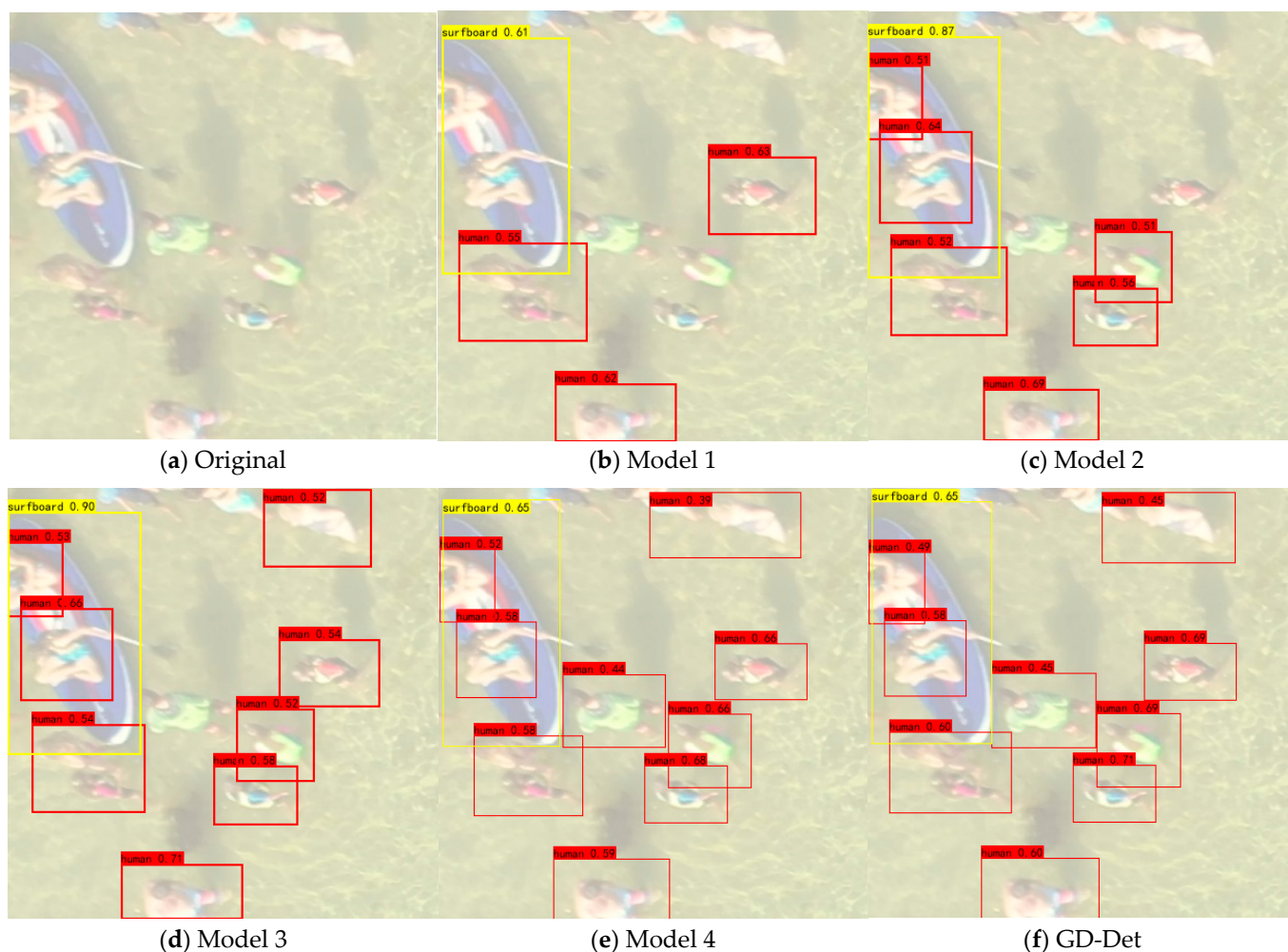


Figure 11. Visualization of Ablation Experiments.

## 5. Conclusions

In this paper, we introduce GD-Det, a low-data object detection model designed to address the challenges of object recognition under low-visibility weather conditions with a small-sample training dataset. GD-Det is built upon re-parameterization and gather-and-distribute mechanisms, and it comprises three main components: a re-parameterized feature extraction module, a gather-and-distribute pyramid module, and a decoupled prediction head. Furthermore, we designed a domain-adaptive training strategy with knowledge transfer and feature–feature alignment.

First, the re-parameterized feature extraction module integrates the re-parameterization mechanism with multi-branch aggregation blocks, enhancing spatial awareness and improving adaptability to objects of varying scales. Second, the gather-and-distribute pyramid module, incorporating gather-and-distribute mechanisms and a self-attention module,

unifies and fuses multi-scale features, strengthening the model's capacity to capture relationships between global and local features. Third, to minimize the interference caused by blurry edges and textures in images captured under adverse weather conditions, a decoupled detection head is introduced. This separates the classification and regression tasks, enabling them to function independently and enhancing feature extraction. In addition, a domain-adaptive training strategy is introduced to overcome the challenges of training with datasets collected under adverse weather conditions. All the experiments conducted on our constructed dataset mini-AFO-fog validate the effectiveness of our model GD-Det, which achieved a detection accuracy of 84% with low-data by addressing the low detection accuracy caused by the image qualities in adverse weather scenarios. It provides a novel solution to address the challenges of multi-scale objects and sample scarcity in remote sensing images.

At the same time, our model has some limitations. The mini-AFO-fog dataset mentioned in this paper is constructed based on publicly available datasets. Although it considers both normal and adverse weather conditions, the scale and diversity of the dataset remain limited. A small-scale dataset may lead to insufficient model generalization, making it difficult to handle more complex and diverse real-world scenarios. Additionally, the ability to generalize to natural images is dependent on the image samples.

**Author Contributions:** Methodology, L.Z. and R.S.; software, S.J.; validation, G.W.; formal analysis, N.Z.; data curation, C.W.; writing—original draft, L.Z.; writing—review and editing, R.S. and S.J.; visualization, G.W.; supervision, L.Z.; funding acquisition, R.S. and L.Z. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported by the National Key R&D Program of China (No. 2023YFC3006500).

**Data Availability Statement:** No new data were created or analyzed in this study. Data sharing is not applicable to this article.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## References

1. Martinez-Esteso, J.; Castellanos, F.; Calvo-Zaragoza, J.; Gallego, A. Maritime Search and Rescue Missions with Aerial Images: A Survey. *arXiv* **2024**, arXiv:2411.07649.
2. Qiu, T.; Zeng, W.; Chen, C. FFDEFA-YOLO: An Enhanced YOLOv8 Model for UAV-Assisted Maritime Rescue Missions. In Proceedings of the 2024 9th International Symposium on Computer and Information Processing Technology (ISCRIPT), Xi'an, China, 24–26 May 2024; pp. 60–63.
3. Kapoor, C.; Warriar, A.; Singh, M.; Narang, P.; Puppala, H.; Rallapalli, S. Fast and lightweight UAV-based road image enhancement under multiple Low-Visibility conditions. In Proceedings of the 2023 IEEE International Conference on Pervasive Computing and Communications (PerCom) Workshops and Other Affiliated Events, Atlanta, GA, USA, 13–17 March 2023; pp. 154–159.
4. Duan, R.; Wu, B.; Zhou, H.; Zhou, H.; He, Z.; Xiao, Z.; Fu, C. E3-Net: Event-Guided Edge-Enhancement Network for UAV-based Crack Detection. In Proceedings of the 2024 International Conference on Advanced Robotics and Mechatronics (ICARM), Tokyo, Japan, 6–8 July 2024; pp. 272–277.
5. Zhao, C.; Liu, R.; Qu, J.; Gao, R. Deep learning-based object detection in maritime unmanned aerial vehicle imagery: Review and experimental comparisons. *Eng. Appl. Artif. Intell.* **2024**, *128*, 107513. [[CrossRef](#)]
6. Liu, H.; Galindo, M.; Xie, H.; Wong, L.; Shuai, H.; Li, Y.; Cheng, W. Lightweight Deep Learning for Resource-Constrained Environments: A Survey. *ACM Comput. Surv.* **2024**, *56*, 1–42. [[CrossRef](#)]
7. Liang, J.; Wu, Y.; Qin, Y.; Wang, H.; Li, X.; Peng, Y.; Xie, X. CPDet: Circle-Permutation-Aware Object Detection for Heat Exchanger Cleaning. *Appl. Sci.* **2024**, *14*, 9115. [[CrossRef](#)]
8. Wang, S.; Chen, D.; Xiang, J.; Zhang, C. A Deep-Learning-Based Detection Method for Small Target Tomato Pests in Insect Traps. *Agronomy* **2024**, *14*, 2887. [[CrossRef](#)]
9. Du, L.; Zhang, R.; Wang, X. Overview of two-stage object detection algorithms. *J. Phys. Conf. Ser.* **2020**, *1544*, 012033. [[CrossRef](#)]
10. He, K.; Zhang, X.; Ren, S.; Sun, J. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 1904–1916. [[CrossRef](#)] [[PubMed](#)]

11. Girshick, R. Fast r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
12. 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.* **2016**, *39*, 1137–1149. [[CrossRef](#)]
13. Liu, S.; Qi, L.; Qin, H.; Shi, J.; Jia, J. Path aggregation network for instance segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 8759–8768.
14. Cai, Z.; Vasconcelos, N. Cascade r-cnn: Delving into high quality object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2018, Salt Lake City, UT, USA, 18–23 June 2018; pp. 6154–6162.
15. Zhao, Q.; Sheng, T.; Wang, Y.; Tang, Z.; Chen, Y.; Cai, L.; Ling, H. M2det: A single-shot object detector based on multi-level feature pyramid network. In Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; pp. 9259–9266.
16. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, VA, USA, 27–30 June 2016; pp. 779–788.
17. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.-Y.; Berg, A.C. SSD: Single Shot Multibox Detector. In Proceedings of the Computer Vision—ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016; pp. 21–37.
18. Law, H.; Deng, J. Cornernet: Detecting objects as paired keypoints. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 734–750.
19. Zhou, X.; Wang, D.; Krähenbühl, P. Objects as points. *arXiv* **2019**, arXiv:1904.07850.
20. Tian, Z.; Shen, C.; Chen, H.; He, T. Fcos: Fully convolutional one-stage object detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 9627–9636.
21. Kong, T.; Sun, F.; Liu, H.; Jiang, Y.; Shi, J. Foveabox: Beyond anchor-based object detector. *arXiv* **2019**, arXiv:1904.0379729.
22. Bousetouane, F.; Morris, B. Fast CNN surveillance pipeline for fine-grained vessel classification and detection in maritime scenarios. In Proceedings of the 13th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), Colorado Springs, CO, USA, 23–26 August 2016; pp. 242–248.
23. Yang, Y.; Chen, T.; Zhou, G. Domain Adaptive Multitask Model for Object Detection in Foggy Weather Conditions. In Proceedings of the 2023 China Automation Congress (CAC), Chongqing, China, 17–19 November 2023; pp. 7280–7285.
24. Ma, J.; Lin, M.; Zhou, G.; Jia, Z. Joint Image Restoration For Domain Adaptive Object Detection In Foggy Weather Condition. In Proceedings of the International Conference on Image Processing (ICIP), Abu Dhabi, United Arab Emirates, 27–30 October 2024; pp. 542–548.
25. Zhang, L.; Zhang, N.; Shi, R.; Wang, G.; Xu, Y.; Chen, Z. Sg-det: Shuffle-ghostnetbased detector for real-time maritime object detection in uav images. *Remote Sens.* **2023**, *15*, 3365. [[CrossRef](#)]
26. Yue, T.; Yang, Y.; Niu, J.M. A Light-weight Ship Detection and Recognition Method Based on YOLOv4. In Proceedings of the 2021 4th International Conference on Advanced Electronic Materials, Computers and Software Engineering (AEMCSE), Changsha, China, 26–28 March 2021; pp. 661–670.
27. Kozlov, A.; Andronov, V.; Gritsenko, Y. Lightweight network architecture for real-time action recognition. In Proceedings of the 35th Annual ACM Symposium on Applied Computing, Brno, Czech Republic, 30 March–3 April 2020; pp. 2074–2080.
28. Zhang, X.; Zhou, X.; Lin, M.; Sun, J. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 6848–6856.
29. Mehta, S.; Rastegari, M.; Caspi, A.; Hajishirziet, H. Espnet: Efficient spatial pyramid of dilated convolutions for semantic segmentation. In Proceedings of the european conference on computer vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 552–568.
30. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv* **2017**, arXiv:1704.04861.
31. Ma, N.; Zhang, X.; Zheng, H.T.; Sun, J. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 116–131.
32. Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 3431–3440.
33. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.C. Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4510–4520.
34. Howard, A.; Sandler, M.; Chu, G.; Chen, L.C.; Chen, B.; Tan, M.; Wang, W.; Zhu, Y.; Pang, R.; Vasudevan, V.; et al. Searching for mobilenetv3. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 1314–1324.

35. Zheng, M.; Sun, L.; Dong, J.; Pan, J. SMFANet: A Lightweight Self-Modulation Feature Aggregation Network for Efficient Image Super-Resolution. In Proceedings of the European Conference on Computer Vision (ECCV), Milan, Italy, 29 September–4 October 2024; pp. 359–375.
36. Potje, G.; Cadar, F.; Araujo, A.; Martins, R.; Nascimento, E. XFeat: Accelerated Features for Lightweight Image Matching. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 17–21 June 2024; pp. 2682–2691.
37. Zhang, H.; Gong, Y.; Yao, F.; Zhang, H. Research on real-time detection algorithm for pedestrian and vehicle in foggy weather based on lightweight XM-YOLOvT. *IEEE Access* **2024**, *12*, 7864–7883. [\[CrossRef\]](#)
38. Hou, W.; Cui, K.; Yan, J.; Chang, J. Lightweight Foggy Weather Detection Algorithm Based on YOLOv5. In Proceedings of the 2024 4th International Conference on Machine Learning and Intelligent Systems Engineering (MLISE), Zhuhai, China, 28–30 June 2024; pp. 338–341.
39. Zhang, L.; Xu, M.; Wang, G.; Shi, R.; Xu, Y.; Yan, R. SiameseNet Based Fine-Grained Semantic Change Detection for High Resolution Remote Sensing Images. *Remote Sens.* **2023**, *15*, 5631. [\[CrossRef\]](#)
40. Ding, X.; Zhang, X.; Ma, N.; Han, J.; Ding, G.; Sun, J. Repvgg: Making vgg-style convnets great again. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 13733–13742.
41. Ding, X.; Zhang, X.; Han, J.; Ding, G. Diverse branch block: Building a convolution as an inception-like unit. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Virtual Event, 19–25 June 2021; pp. 10886–10895.
42. Kang, M.; Ting, C.; Ting, F. RCS-YOLO: A fast and high-accuracy object detector for brain tumor detection. In Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention, Vancouver, BC, Canada, 8–12 October 2023; pp. 600–610.
43. Zhang, Z.; Dong, Z.; Xu, W.; Han, J. Re-Parameterization of Lightweight Transformer for On-Device Speech Emotion Recognition. *IEEE Internet Things J.* **2024**. [\[CrossRef\]](#)
44. Lin, T.Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature pyramid networks for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2117–2125.
45. Wang, C.; He, W.; Nie, Y.; Guo, J.; Liu, C.; Wang, Y.; Han, K. Gold-YOLO: Efficient object detector via gather-and-distribute mechanism. *Adv. Neural Inf. Process. Syst.* **2024**, *36*. [\[CrossRef\]](#)
46. Deng, X.; Mao, Y.; Deng, D. Deep Convolutional Neural Network Algorithm Based on Im2Col and Non-Local Mean Filter. In Proceedings of the 2023 4th International Seminar on Artificial Intelligence, Networking and Information Technology (AINIT), Nanjing, China, 16–18 June 2023; pp. 6–10.
47. Gasienica-Jozkow, J.; Knapik, M.; Cyganek, B. An ensemble deep learning method with optimized weights for drone-based water rescue and surveillance. *Integr. Comput.-Aided Eng.* **2021**, *28*, 221–235. [\[CrossRef\]](#)
48. Zhang, N.; Zhang, L.; Cheng, Z. Towards simulating foggy and hazy images and evaluating their authenticity. In Proceedings of the Neural Information Processing: 24th International Conference (ICONIP), Guangzhou, China, 14–18 November 2017; pp. 405–415.

**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.