



Article

CoDerainNet: Collaborative Deraining Network for Drone-view Object Detection in Rainy Weather Conditions

Yue Xi ¹, Wenjing Jia ² , Qiguang Miao ^{3,*} , Junmei Feng ¹, Xiangzeng Liu ³ and Fei Li ³¹ Guangzhou Institute of Technology, Xidian University, Guangzhou 510555, China² Global Big Data Technologies Centre, University of Technology Sydney, Ultimo, NSW 2007, Australia³ School of Computer Science and Technology, Xidian University, Xi'an 710071, China

* Correspondence: qgmiao@xidian.edu.cn

Abstract: Benefiting from the advances in object detection in remote sensing, detecting objects in images captured by drones has achieved promising performance in recent years. However, drone-view object detection in rainy weather conditions (Rainy DroneDet) remains a challenge, as small-sized objects blurred by rain streaks offer a little valuable information for robust detection. In this paper, we propose a Collaborative Deraining Network called “CoDerainNet”, which simultaneously and interactively trains a deraining subnetwork and a droneDet subnetwork to improve the accuracy of Rainy DroneDet. Furthermore, we propose a Collaborative Teaching paradigm called “ColTeaching”, which leverages rain-free features extracted by the Deraining Subnetwork and teaches the DroneDet Subnetwork such features, to remove rain-specific interference in features for DroneDet. Due to the lack of an existing dataset for Rainy DroneDet, we built three drone datasets, including two synthetic datasets, namely RainVisdrone and RainUAVDT, and one real drone dataset, called RainDrone. Extensive experiment results on the three rainy datasets show that CoDerainNet can significantly reduce the computational costs of state-of-the-art (SOTA) object detectors while maintaining detection performance comparable to these SOTA models.

Keywords: drone-view object detection; image deraining; collaborative teaching

Citation: Xi, Y.; Jia, W.; Miao, Q.; Feng, J.; Liu, X.; Li, F. CoDerainNet: Collaborative Deraining Network for Drone-view Object Detection in Rainy Weather Conditions. *Remote Sens.* **2023**, *15*, 1487. <https://doi.org/10.3390/rs15061487>

Academic Editor: Eufemia Tarantino

Received: 2 February 2023
Revised: 24 February 2023
Accepted: 26 February 2023
Published: 7 March 2023



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

1. Introduction

Drones have attracted much attention recently due to their rapid and cost-effective deployment [1]. Drone-view object detection (DroneDet) aims to locate and classify objects in images captured by drones, which is one of the most crucial algorithms deployed on drones for environmental perception. Recently, a few object detectors for DroneDet [2–4] have been proposed to boost detection performance. Their detection accuracy decreases enormously in rainy weather, which is one of the most common weather conditions, although they have achieved impressive performance in favorable weather conditions.

Rain contains countless rain streaks, which have different density levels. These rain streaks block some of the light reflected by objects, thus decreasing the contrast between objects and the background in an image. A widely used rain model is the additive composite model [5,6], which is written as follows:

$$r = c + s \quad (1)$$

where r is an image degraded by rain streaks, c is the corresponding rain-free and clean image, and s denotes rain streaks, which can be viewed as additive noise. The noise s degrades features extracted for DroneDet in rainy weather conditions, resulting in poor detection performance.

We explain the reasons for the poor detection performance from a probabilistic perspective. First, we define some notations for further analysis. Denote a rain-free (source) domain as $\mathcal{D} = \{C, P(C)\}$ (indicated as the black circles in Figure 1). \mathcal{D} consists of clean

images collected under favorable weather conditions, where \mathbf{C} is a feature space, $P(\mathbf{C})$ is a marginal probability distribution, and $\mathbf{C} = \{c_1, \dots, c_n\} \in \mathbf{C}$. Denote a rainy (target) domain as $\mathcal{D}^r = \{\mathbf{R}, P^r(R)\}$ (indicated as the pink circles in Figure 1). \mathcal{D}^r consists of degraded images collected under rainy weather conditions, where \mathbf{R} is another feature space, $P^r(R)$ is a marginal probability distribution, and $R = \{r_1, \dots, r_n\} \in \mathbf{R}$. Let the task of drone-view object detection under rainy weather conditions (Rainy DroneDet) be $\mathcal{T}^r = \{\mathbf{Y}, P(Y|R)\}$, where \mathbf{Y} is a feature space, $P(Y|R)$ is a conditional probability distribution, and $Y = \{y_1, \dots, y_n\} \in \mathbf{Y}$. When a detector that is well trained in \mathcal{D} is utilized to perform DroneDet in \mathcal{D}^r , the detection performance will decrease significantly since $P(Y|\mathbf{C}) \neq P(Y|\mathbf{R})$.

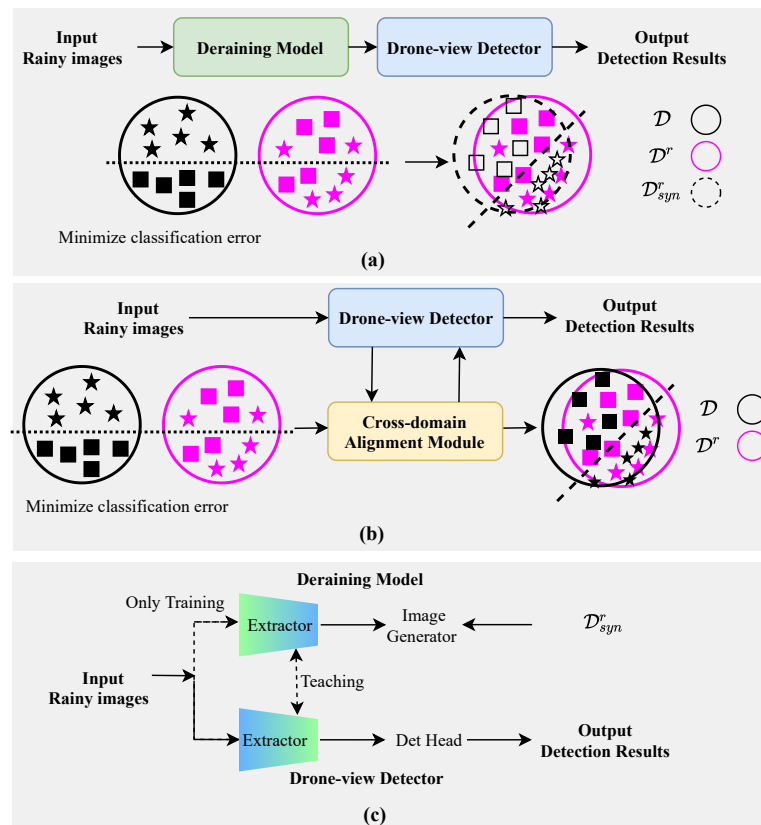


Figure 1. Comparison of the proposed CoDerainNet and existing detectors for Rainy DroneDet. (a) ImDrain-based methods, (b) DA-based methods, (c) our CoDerainNet.

Two types of solutions for the Rainy DroneDet are Image Deraining-based methods (“ImDerain-based”, as illustrated in Figure 1a) and Domain Adaptation-based methods (“DA-based”, as illustrated in Figure 1b). The ImDerain-based methods [7–9] generally consist of two stages: image deraining and object detection. However, they adopt a multi-stage and progressively deraining model [10] to obtain rain-free images, resulting in huge computation costs. Deploying these ImDerain-based methods on drones is infeasible due to the very limited onboard computing resources.

Essentially, these ImDerain-based methods attempt to build a synthetic rainy domain to mitigate the domain gap between \mathcal{D} and \mathcal{D}^r , as shown in Figure 1a. Let the synthetic rainy domain be $\mathcal{D}_{syn}^r = \{\mathbf{R}_{syn}, P^r(R_{syn})\}$, where \mathbf{R}_{syn} is another feature space, and $P^r(R_{syn})$ is a marginal probability distribution, $R_{syn} = \{r_{syn}^1, \dots, r_{syn}^n\} \in \mathbf{R}_{syn}$, and r_{syn}^i is synthesized by the combination of c and synthetic rain streaks s_{syn} . However, Wei et al. [11] reported that there was a large difference between s_{syn} and s , such as the direction and density of rain streaks. Instead of building \mathcal{D}_{syn}^r , DA-based methods, [12–14] design a cross-domain alignment module to directly align the two feature spaces, \mathbf{C} and \mathbf{R} , as shown in

Figure 1b. The DA-based methods investigate cross-domain knowledge from a probabilistic perspective, but neglect intrinsic knowledge in the image degradation from \mathcal{D} to \mathcal{D}' .

In this paper, we mitigate the two issues and propose a light image degradation knowledge-transferring network for Rainy DroneDet, called “CoDerainNet”, which is a **C**ollaborative **D**eraining **N**etwork. As shown in Figure 1c, our CoDerainNet includes the Deraining Subnetwork, DroneDet Subnetwork, and a Collaborative Teaching paradigm. CoDerainNet can interactively train the Deraining Subnetwork and DroneDet Subnetwork to improve the Rainy DroneDet performance with limited additional computational cost during inference. Furthermore, we propose a Collaborative Teaching paradigm called “ColTeaching”, which transfers intrinsic degradation knowledge from the Deraining Subnetwork to DroneDet Subnetwork and teaches the DroneDet Subnetwork such knowledge to prevent rain-specific interference in features for DroneDet.

We build three drone-captured datasets due to the scarcity of datasets for Rainy DroneDet. They include two synthetic drone-captured datasets, namely RainVisDrone and RainUAVDT, based on the VisDrone [15] and UAVDT [16] benchmark datasets. Moreover, we create a real drone-captured dataset, “RainDrone”, to verify CoDerainNet’s effectiveness in real rainy scenarios. More details of our RainDrone will be introduced in Section 5.1.

Our main contributions can be summarized as follows.

- (1) We propose CoDerainNet, a light object detector for Rainy DroneDet that can interactively train the Deraining Subnetwork and DroneDet Subnetwork to improve Rainy DroneDet performance with limited additional computational cost during inference;
- (2) We propose ColTeaching, which transfers intrinsic degradation knowledge from the Deraining Subnetwork to DroneDet Subnetwork to block rain-specific interference in features for Rainy DroneDet. This offers a new solution to the problem of how image restoration techniques can help improve tasks of low-quality image understanding;
- (3) To advance the research on DroneDet under inclement weather, we build three drone-captured datasets, including two synthetic datasets and one real dataset.

We compare CoDerainNet with seven state-of-the-art (SOTA) models to verify its effectiveness and conduct extensive experiments on the three drone-captured datasets. The experiment results show that CoDerainNet can significantly reduce the computational costs of these SOTA object detectors while maintaining detection performance comparable to these SOTA models.

The rest of the paper is organized as follows: Firstly, in Section 2, we review the current development of Rainy DroneDet and provide a summary of related works. Section 3 describes the problem in collaborative deraining learning for Rainy DroneDet. In Section 4, we provide details of CoDerainNet. Section 5 presents the experimental results on CoDerainNet. Section 6 discusses the limitations and presents a discussion of CoDerainNet. Finally, in Section 7, we conclude the paper.

2. Related Work

We review the related works from two aspects: general object detection under rainy weather conditions and DroneDet.

2.1. Object Detection under Rainy Weather Conditions

We describe the major existing solutions to this task in two main directions, i.e., the ImageDe-based and DA-based for general object detection. We provide a summary of their respective advantages and disadvantages in Table 1.

Table 1. Summary of the advantages and disadvantages of object detection under rainy weather conditions.

Methods	Advantages	Disadvantages
ImageDe-based methods [17–19]	Intuitive idea; adopts the deraining algorithms as a pre-processing module to improve the visibility of the input images.	The pre-processing module increases the complexity of the original detectors. New rainy images need to be annotated to supervise the deraining process.
DA-based methods [13,14,20]	Has no need to annotate new rainy images because they belong to unsupervised learning.	Require a new alignment module for every new domain, which limits their generalization.

2.1.1. ImageDe-Based Object Detection

ImageDe-based object detection [17–19] improves the visibility of input using existing deraining algorithms [17,18] as a pre-processing module, and then conducts object recognition. Early on, the two-stage fashion is very inefficient because an image deraining module in the first stage and an object detection module in the second stage are optimized separately. The image deraining module is not optimized for the task of object recognition but for human perception. Thus, improving image visibility with the image deraining module does not necessarily benefit object detection performance. Furthermore, a unified fashion has been proposed to bridge the goals of image deraining and object detection. Very recently, Liu et al. [19] designed a fully differentiable image restoration module to recover the latent content for the sequential object recognition module. In this model, the two modules can be trained in an end-to-end fashion.

The use of image deraining modules significantly increases the complexity of the original detectors, making it infeasible to deploy them on drones with limited onboard computing resources. Moreover, annotating drone images is more time-consuming and laborious compared to annotating natural scene images, due to the presence of a large number of small objects in drone-captured images.

2.1.2. DA-Based Object Detection

DA-based object detection [13,14,20] bridges the domain gap between images collected in normal (source domain) and rainy weather (target domain) conditions by discovering domain invariant feature representations. Following the domain adaptation paradigm, firstly, samples from the source domain are used to train a detector. Secondly, domain invariant features for the target domain are learned by a feature alignment module in the detector. A multi-adversarial Faster-RCNN detector [20] was proposed for addressing the problem of domain adaptation between normal and rainy weather conditions. The multi-adversarial detector adopted a hierarchical feature alignment module for layer-wise domain invariant features. Sindagi et al. [13] proposed a prior-based domain adversarial framework to adapt existing detectors to inclement weather conditions. In this framework, a prior-adversarial loss was defined to reduce the rainy weather-specific information. The novel loss was used to improve the detection performance under rainy weather. Very recently, for rainy scene segmentation, Lee et al. proposed a segmentation model FIFO [14], which was insensitive to images' rain-style variation.

From a probabilistic perspective, the DA-based methods focus on the latent relationship between the two different domains aiming to discover the common knowledge between a clean domain and a rainy domain. Whereas our CoDerainNet attempts to explore the process of image degradation from a clean domain to a rainy domain. Moreover, the DA-based methods require a new alignment module for every new domain, which limits their generalization ability.

2.2. DroneDet

DroneDet is an emerging and hot topic in aerial image processing. Aerial images are different from natural images and have unique characteristics. For instance, objects in

these aerial images are generally small, resulting in poor DroneDet performance. There are three types of solutions for DroneDet. Super-resolution-based methods [1,4,21] reconstruct a low-resolution image into a high-resolution one, which contains more details of small objects beneficial for recognizing them. Towards this end, generative adversarial networks, consisting of a generator and a discriminator, are utilized to super-resolve low-resolution images. These methods adopt a multi-stage paradigm, including region proposal, RoIs super-resolution, and object detection, which is inefficient and difficult to optimize in an end-to-end manner. Context-based methods [22,23] establish the relationship between RoIs and their surrounding regions, and integrate it into their original features. It is difficult for drone-captured images to establish such a relationship due to background clutter in drone-captured images. Multi-scale-representation-based methods [24–26] combine spatial information in low-level layers and semantic information in high-level layers for feature enhancement. In recent years, the security aspects of drone communications have started to receive attention [27–29]. Tian et al. [27] proposed an adversarial attack model to conduct adversarial attacks against deep learning-based navigation systems of drones.

Although methods for DroneDet have obtained impressive detection performance, their detection performance decreases significantly in rainy weather conditions. To deal with the issue of Rainy DroneDet, CoDerainNet introduces a subnetwork for image deraining.

3. Problem Definition

We follow Multi-Task Learning (MTL) [30] to define the problem in collaborative deraining learning for Rainy DroneDet.

Definition (MTL). Given n related tasks $\{T_i\}_{i=1}^n$, the goal of MTL is to improve the performance of all or some of the tasks by simultaneously learning the n tasks.

Based on this definition, we can formulate the task of Rainy DroneDet. Recall that $\mathcal{T}^r = \{Y, P(Y|R)\}$ is the task of Rainy DroneDet. Let $\mathcal{T}^{dr} = \{C, P(C|R)\}$ be the task of image deraining. \mathcal{T}^r is trained on the dataset $\{R, Y\}$, which consists of N training samples $\{r_i, y_i\}_{i=1}^N$, where $r_i \in \mathbb{R}^{w \times h \times 3}$ is the i th rainy image and y_i is the image's label for DroneDet. \mathcal{T}^{dr} is trained on the dataset $\{R, C\}$, which consists of M training samples $\{r_i, c_i\}_{i=1}^M$, where $c_i \in \mathbb{R}^{w \times h \times 3}$ is the corresponding rain-free image of r_i for image deraining. Therefore, the problem of the collaborative deraining learning for Rainy DroneDet can be formulated as follows.

Definition (collaborative deraining learning for Rainy DroneDet). Given two tasks, Rainy DroneDet \mathcal{T}^r and image deraining \mathcal{T}^{dr} , the goal of the collaborative deraining learning for Rainy DroneDet is to improve the \mathcal{T}^r performance with limited computational costs during inference by simultaneously optimizing these two tasks.

4. Methodology

4.1. Overview

Figure 2 shows the architecture of the proposed CoDerainNet, which consists of two subnetworks and one training paradigm, i.e., Deraining Subnetwork, DroneDet Subnetwork, or ColTeaching. (1) The Deraining Subnetwork reduces rain streak noise in features by reconstructing rain-free images. (2) The DroneDet Subnetwork detects objects in drone-captured images effectively. (3) ColTeaching incorporates image deraining into the process of Rainy DroneDet by strengthening the interaction between the two subnetworks.

CoDerainNet can significantly improve Rainy DroneDet performance by interactively training the two subnetworks. Firstly, each of them is trained on the dataset $\{r_i, c_i\}_{i=1}^M$ or $\{r_i, y_i\}_{i=1}^N$ so that they obtain optimal image deraining and object detection performance, respectively. Secondly, ColTeaching incorporates image deraining into the process of Rainy DroneDet by allowing interaction between them. Moreover, our proposed CoDerainNet can improve Rainy DroneDet performance with limited additional computational cost, as only the DroneDet Subnetwork needs to be executed during inference. We provide the details of the above two subnetworks and the training paradigm below.

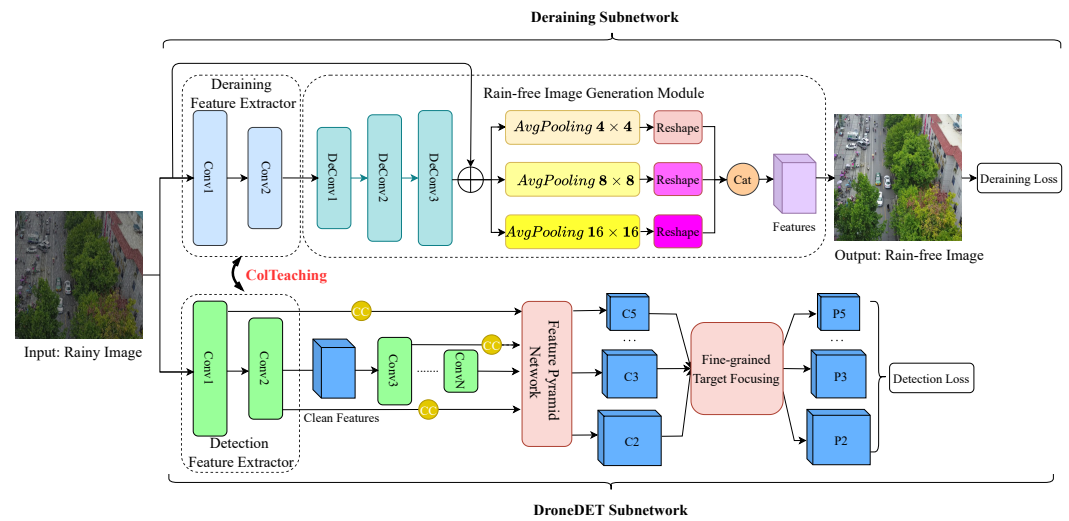


Figure 2. The architecture of CoDerainNet. It consists of two subnetworks and one training paradigm, i.e., Deraining Subnetwork, DroneDet Subnetwork, and ColTeaching. The Deraining Subnetwork is only executed during the training phase.

4.2. Deraining Subnetwork

Rain streaks can reduce the discriminative capacity of different objects' features. In the real world, rain streaks block some of the light reflected by objects, which reduces images' contrast and brightness. In addition, rain streaks appear as noise in images and will reduce signal-to-noise ratio of original images. To alleviate the problem, we proposed a Deraining Subnetwork, which reduces the noise in features by reconstructing rain-free images. Figure 2 shows its architecture, which consists of two modules, namely a deraining feature extractor $\phi_{derain}(\cdot)$ and a rain-free image generation module. To successfully perform the subsequent weight-sharing strategy ColTeaching, $\phi_{derain}(\cdot)$ shares the same structure as the detection feature extractor, which consists of the first two convolution blocks in the DroneDet Subnetwork. Features $\phi_{derain}(R)$ are delivered into the rain-free image generation module for image deraining.

The rain-free image generation module is presented as follows. Firstly, one convolution block $Conv(\cdot)$ and two ResBlock modules $Res(\cdot)$ are utilized to integrate more context information into the extracted features $\phi_{derain}(R)$ for image deraining. Secondly, several deconvolution blocks $DeConv(\cdot)$ are used to up-sample the enriched features to make the resolution of the up-sampled features the same as that of the input. Then, the matched features and their corresponding input are concatenated $Cat(\cdot, \cdot)$ and pass through the pyramid enhancement block [31] $Pyramid(\cdot)$ to reconstruct the rain-free images. $Pyramid(\cdot)$ is responsible for improving the features' representational power through multi-scale learning and multiple convolutional blocks with different receptive fields. At last, the enhanced features pass through a 3×3 convolution layer $Conv_{3 \times 3}(\cdot)$ to obtain the final rain-free images. Mathematically, the generated rain-free images $C = \{c_1, c_2, \dots, c_N\}$ can be formulated as follows:

$$C = Conv_{3 \times 3}(Pyramid(Cat(feats_{noise}, R))) \quad (2)$$

where

$$feats_{noise} = DeConv(Res(Conv(\phi_{derain}(R)))). \quad (3)$$

The loss function of the Deraining Subnetwork \mathcal{L}_{DR} is written as follows:

$$\mathcal{L}_{DR} = \frac{1}{N} \sum_{i=1}^N \|c_i - c_i^{GT}\|_2, \quad (4)$$

where N denotes the batch size and $\|\cdot\|_2$ presents the $L2$ norm, and c_i and c_i^{GT} represent the i th generated image and the corresponding ground truth, respectively. In this way, the clean features $\phi_{derain}(R)$, where noise is removed by image deraining, are beneficial to Rainy DroneDet.

The pseudo-code of image deraining with the Deraining Subnetwork is summarized in Algorithm 1.

Algorithm 1 The pipeline of image deraining with the Deraining Subnetwork.

Input: Rainy images $R = \{r_1, \dots, r_n\} \in \mathbf{R}$.

Output: Rain-free images $C = \{c_1, \dots, c_n\} \in \mathbf{C}$

```

1: begin
2:   for  $i \in [0, n]$  do
3:      $\phi_{derain}(r_i)$ ;
4:      $feat_{cont} = Res(Conv(\phi_{derain}(R)))$ ;
5:      $feat_{noise} = DeConv(feat_{cont})$ ;
6:      $feat_{cat} = Cat(feat_{noise}, R)$ ;
7:      $feat_{enh} = Pyramid(feat_{cat})$ ;
8:      $c_i = Conv_{3 \times 3}(feat_{enh})$ ;
9:      $loss = \|c_i - c_i^{GT}\|_2$ ;
10:    Loss Back-forward Propagation;
11:  end
12: end

```

4.3. DroneDet Subnetwork

The DroneDet Subnetwork adopts Feature Pyramid Networks (FPN) [32] to detect small objects effectively. Figure 2 presents its architecture, which includes four modules. The first module is a backbone, which transforms the input into a certain feature representation. The second module is an FPN module, which utilizes convolution kernel filters with different strides to build several pyramidal feature hierarchies. It extracts multi-scale feature representations by combining the rich semantic and detailed position information from different layers of the backbone. The third module is a fine-grained target-focusing module, which refines the multi-scale feature representations. The last module, “Head”, is used to predict positions and categories of objects. A module for context collection is also integrated into the second module to improve the expressive power of small-sized objects’ features.

Context collector [26] is a module specially designed for DroneDet to collect local and global context information to boost small object detection performance. It includes three branches: a 1×1 convolutional layer, a dilated convolutions, and a global average pooling layer. In the first branch, the 1×1 convolutional layer ψ_i is utilized to decrease the channel number of input feature C_i . In the second branch, a few 3×3 convolutional filters v_i^k with the atrous rate $k \in (1, 2, \dots, N)$ are adopted to build local contextual features. In the last branch, the global average pooling layer ϕ_i is used for a collection of global contextual information at image level. The final features are obtained by concatenating features extracted from the three branches. The series of operations is written as follows:

$$F_i(C_i) = Con(\{v_i^k(C_i)\}_{k=1}^N, \psi_i(C_i), \phi_i(C_i)). \quad (5)$$

where Con is the concatenation operator.

The fine-grained target-focusing module [26] is introduced to achieve a better performance of small object detection. This module further enhances the multi-scale features by aggregating fine-grained objects’ sub-parts with a special focus on small objects. Figure 3 presents the details of its structure, including FiFA and TFB. Specifically, FiFA is a fine-grained feature aggregation block, which adaptively aggregates sub-regions from

multi-scale features. TFB is a target-focusing block, which focuses the attention on RoIs to suppress background noise.

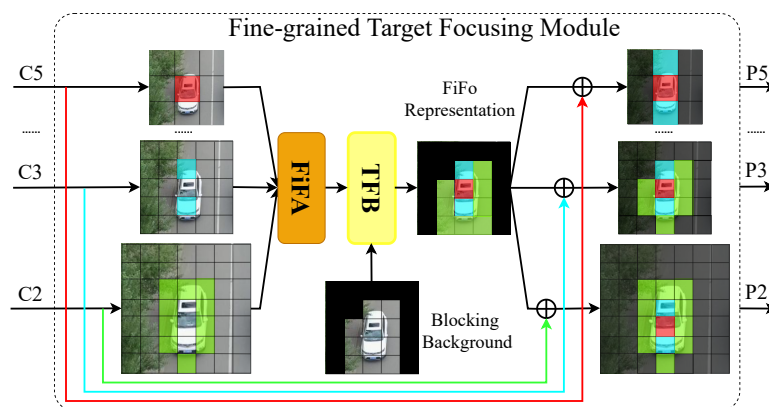


Figure 3. The details of the Fine-grained Target-focusing module.

A regression loss for objects’ locations and a classification loss for objects’ categories are used to optimize DroneDet Subnetwork. The regression loss *GIoU* [33] is used to regress predicted bounding boxes to solve vanishing gradients of *IoU* loss for non-overlapping case. The *GIoU* loss is written as follows:

$$GIoU = IoU - \frac{|C - (A \cup B)|}{|C|}, \tag{6}$$

where

$$IoU = \frac{|A \cap B|}{|A \cup B|}. \tag{7}$$

Here, *A* is a position of a predicted bounding box, *B* is a position of a ground truth bounding box, and *C* is the smallest convex set of $|A \cup B|$.

The binary cross entropy (BCE) loss is used to predict objects’ categories. The BCE loss is written as follows:

$$\mathcal{L}_{BCE} = -(y \cdot \log(\hat{y}) + (1 - y) \cdot \log(1 - \hat{y})), \tag{8}$$

where \hat{y} is a predicted probability of a sample, and *y* is a label of the sample.

4.4. Collaborative Teaching Paradigm (ColTeaching)

We proposed ColTeaching to perform interactions between the above two subnetworks. ColTeaching is a simple but effective joint optimization paradigm, which attempts to incorporate image deraining into the process of Rainy DroneDet. Figure 4 illustrates the process of ColTeaching. Firstly, Deraining Subnetwork and DroneDet Subnetwork are pre-trained with their task-related annotations, namely the object’s position and categories for detection and clean images for deraining. Secondly, given the two pre-trained subnetworks, the detection feature extractor and the deraining feature extractor exchange their weights. Finally, the two subnetworks are trained with their respective task for *n* epochs and then exchange weights of their feature extractors.

A reasonable explanation for why ColTeaching can be effective is as follows. Cleaner and semantically more meaningful features are built through ColTeaching and used to teach the two subnetworks, allowing them to benefit each other. The clean features extracted by Deraining Subnetwork are viewed as intrinsic degradation knowledge and used to teach DroneDet Subnetwork to block rain-specific interference in features for DroneDet. Conversely, the semantic features extracted by DroneDet Subnetwork are viewed as helpful knowledge and used to teach Deraining Subnetwork to generate the rain-free image.

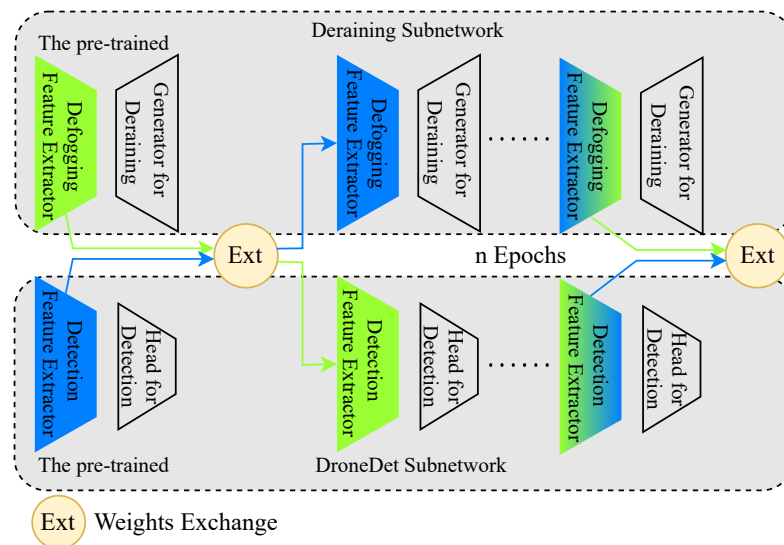


Figure 4. The process of our ColTeaching.

5. Experiments

We evaluated our CoDerainNet on the three datasets collected on drones under rainy weather conditions, and compared it with seven SOTA methods to verify its effectiveness. We first introduce experimental datasets and compare models below.

5.1. Datasets and Models

No benchmark datasets have been built for Rainy DroneDet. According to the very recent review [15] for drone-based vision, only two public datasets have been built for DroneDet, i.e., VisDrone [15] and UAVDT [16]. However, these two datasets do not contain any images under inclement weather conditions, especially rainy weather conditions. Therefore, we build two synthetic drone-captured datasets, i.e., RainVisDrone and RainUAVDT, and create a real drone-captured dataset, i.e., RainDrone. Table 2 shows the overview of our drone-captured dataset.

We synthesized rainy images based on the rain synthesis process in [34]. Rainy images from RainVisDrone and RainUAVDT are synthesized with three different rain-density levels, namely light, medium, and heavy level, as was in SFA-Net [35]. The noise level is introduced to adjust an image’s rain density. Light, medium, and heavy rain conditions correspond to the noise level 5~40%, 40~70%, and 70~95%, respectively. Figure 5a shows some samples of the synthesized images at three rain-density levels.

Table 2. Overview of our drone-captured rainy image datasets.

Dataset	Rain Density	#Images	Real or Synthetic
RainVisDrone	light	7019	synthetic
	medium	7019	synthetic
	heavy	7019	synthetic
RainUAVDT	light	38,327	synthetic
	medium	38,327	synthetic
	heavy	38,327	synthetic
RainDrone	N/A	300	real

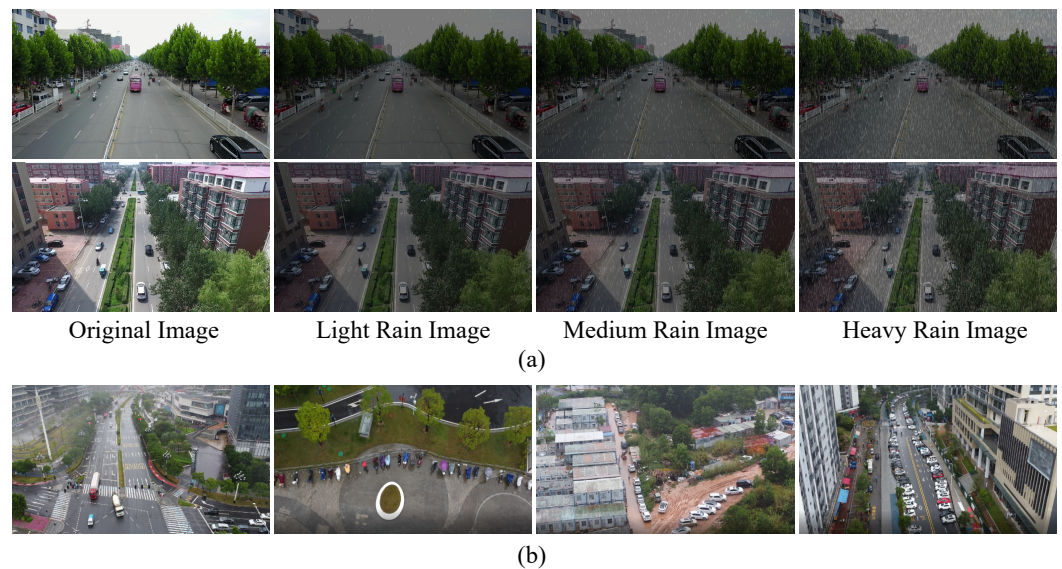


Figure 5. Samples of images from our datasets. (a) Synthetic rainy images from RainVisDrone and RainUAVDT. (b) Real rainy images from RainDrone.

- (1) RainVisDrone: Our RainVisDrone dataset has 19,413 synthetic images for training and 1644 synthetic images for testing. The image resolution is about 2000×1500 pixels. Images have ten categories, i.e., pedestrian, person, car, van, bus, truck, motor, bicycle, awning tricycle, and tricycle;
- (2) RainUAVDT: Our RainUAVDT dataset has 69,774 synthetic images for training and 45,207 synthetic images for testing. The image resolution is about 1080×540 pixels. Images have three categories, i.e., bus, truck, and car;
- (3) RainDrone: All images from RainDrone are captured by a drone platform DJI MINI 2. They mainly include scenarios under rainy weather conditions. RainDrone has 300 real rainy images for model inference. The resolution of images is about 1080×540 pixels. RainDrone has the same categories as RainVisdrone, including pedestrian, person, car, van, etc. Figure 5b shows some samples of real rainy images from RainDrone.

According to Section 2, we designed three groups of experiments to comprehensively compare CoDerainNet with object detection models. In the first group, CoDerainNet is compared with two current SOTA object detection models, namely YOLOv5 [36] and YOLOv7 [37]. In the second group, CoDerainNet is compared with the ImDerain-based methods, which are combination models of MPRNet [7], PReNet [8], and each of the two SOTA detectors: (1) MPRNet and YOLOv5, called “MPRNet-YOLOv5”; (2) PReNet and YOLOv5, called “PReNet-YOLOv5”; (3) MPRNet and YOLOv7, called “MPRNet-YOLOv7”; (4) PReNet and YOLOv7, called “PReNet-YOLOv7”. In the third group, CoDerainNet is compared with the DA-based method AdaptiveTeacher [38].

5.2. Implementation Details and Evaluation Metrics

5.2.1. Implementation Details

We conducted all the implements with PyTorch 1.8.1 on a server with NVIDIA RTX3090 GPU. We used part of the pre-trained model from YOLOv5 [36] to save training time. The Adam optimizer is used to train CoDerainNet because it can better deal with the issue of sparse gradients and converge faster than the standard SGD optimizer. The learning rate with the cosine learning rate schedule is set to 3×10^{-4} . The long side of the input image to CoDerainNet is 1536 pixels, the same as in TPH-YOLOv5 [39].

5.2.2. Evaluation Metrics

We adopted Average Precision (AP) and mean Average Precision (mAP), the same as in PASCAL VOC [40], to evaluate the detection performance of CoDerainNet. AP is defined as follows:

$$AP = \int_0^1 P(R)dR, \quad (9)$$

where R is Recall and P is Precision, and $P(R)$ is the curve composed of R and P . mAP is defined as follows:

$$mAP = \frac{1}{N} \sum_{i=1}^N AP_i, \quad (10)$$

where N is the number of categories. AP is averaged on ten Intersection over Union (IoU) values of $[0.50 : 0.05 : 0.95]$. AP_{50} and AP_{75} are calculated at the single IoU of 0.5 and 0.75, respectively.

5.3. Ablation Studies

We carried out ablation studies on RainVisDrone to verify the effectiveness of each component in CoDerainNet. We removed the Deraining Subnetwork from CoDerainNet as the baseline model.

5.3.1. Effectiveness of Our ColTeaching

We aimed to verify that CoDerainNet can perform weights exchange by the ColTeaching paradigm, to effectively improve the detection accuracy for DroneDet. For the interval of exchanging weights, we assigned weights of the Deraining Subnetwork to the DroneDet Subnetwork every 50 epochs; we assigned weights of the DroneDet Subnetwork to the Deraining Subnetwork every 600 epochs. We set the input of image size as 640×640 instead of large input 1336×1336 to reduce the training time of CoDerainNet.

Figures 6 and 7 show the learning process of both the DroneDet and Deraining Subnetwork optimized by our ColTeaching paradigm, respectively. In Figure 6, we can see that the detection accuracy of the DroneDet Subnetwork significantly drops and then gradually increases every 50 epochs. As for Box_loss and Cls_loss, the two losses significantly boost and gradually decrease every 50 epochs. This result shows that the DroneDet Subnetwork can still converge effectively with ColTeaching. Figure 7 presents that the PSNR value of reconstructed rain-free images significantly drops and then gradually increases every 600 epochs. The \mathcal{L}_{DR} significantly boosts and gradually decreases every 600 epochs. This result suggests that the Deraining Subnetwork can also converge effectively with ColTeaching. In addition, Table 3 presents that the AP_{50} of “Baseline w \mathcal{L}_{DR} ” increased by 2.18%. The results verify that the ColTeaching paradigm can effectively boost the DroneDet performance. With the ColTeaching paradigm, clean features extracted by the Deraining Subnetwork from rainy inputs for image deraining can be shared to learn better detection features in the DroneDet Subnetwork.

Table 3. Effectiveness of the Deraining Subnetwork.

Method	AP_{50} [%]
Baseline	37.97
Baseline w Conv_3	39.01
Baseline w Conv_2	39.76
Baseline w \mathcal{L}_{DR} (Baseline w Conv_1)	40.15

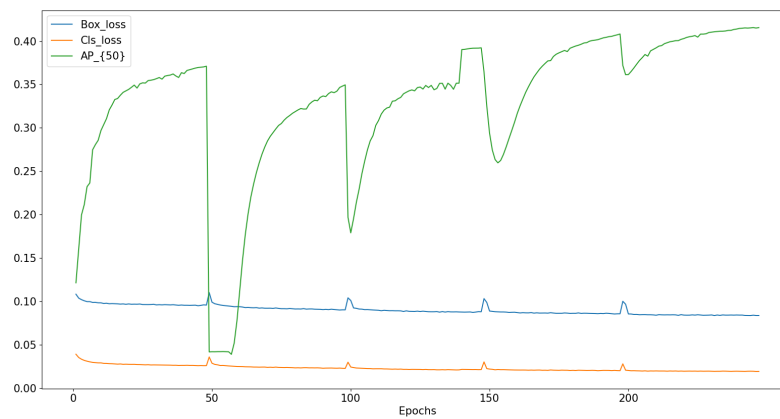


Figure 6. Curves of \mathcal{L}_{DR} on the training set for medium rain.

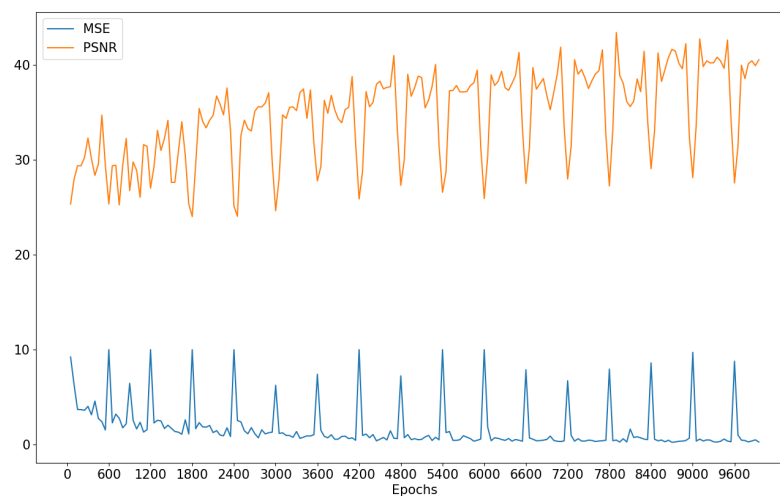


Figure 7. Curves of \mathcal{L}_{DR} and PSNR with ColTeaching.

5.3.2. Comparison of Different Implementations of Our ColTeaching

We further investigated three implementations of ColTeaching to select an optimal configuration of the deraining feature extractor. The first implementation, called “Baseline w Conv_1”, adopts the first two convolution blocks of the detection subnetwork as the deraining feature extractor. The second implementation, called “Baseline w Conv_2”, adopts one more convolution block to learn deeper features. We continue exploring the deeper features as the deraining feature extractor with the final implementation, called “Baseline w Conv_3”, which uses all convolution blocks. Table 3 compares the DroneDet accuracy of CoDerainNet with different convolution blocks. The experimental results demonstrate that “Baseline w Conv_1” can obtain the best DroneDet performance.

To interpret the improved performance, note that, compared with Conv_2 and Conv_3, features extracted by Conv_1 have the maximum resolution and hence contain the richest spatial details. Deraining features extracted with the deeper convolutional blocks (Conv_2 or Conv_3) contain less amount of spatial details due to their low resolution. The visibility enhancement for input images of these features is limited, which leads to the poor quality of features for the DroneDet Subnetwork.

5.3.3. Learning Curves of the Proposed Deraining Subnetwork

We aimed to verify that the proposed Deraining Subnetwork can converge stably during training and effectively remove rain streaks in a rainy image. The dataset consists of images selected randomly from our RainVisDrone for image deraining. It has 432 training pairs of rainy and clean images and another 20 pairs for evaluation, which is the same as DnCNN [41]. The resolution of the training patches randomly cropped from their

original images is 48×48 pixels. In addition, two metrics, structure similarity index [42] (SSIM) and peak signal-to-noise ratio [43] (PSNR), are used to evaluate the reconstructed images' quality.

Figure 8 presents the \mathcal{L}_{DR} loss curves of the Deraining Subnetwork on the training set for light, medium, and heavy rain, respectively. A value on the horizontal axis represents 50 epochs. We observe that the three loss curves continuously decrease to stable points after 6000 (120×50) epochs. Therefore, our Deraining Subnetwork can converge stably for light, medium, and heavy rainy datasets. Furthermore, Figures 9 and 10 show that with the increase in training epochs, the PSNR and SSIM curves of the reconstructed images increase. Similarly, the two metrics values tend to be stable after 6000 (120×50) epochs. Table 4 compares the average values for PSNR and SSIM of images reconstructed from images with different rain density. We can observe that our Deraining Subnetwork can still achieve 40.10 dB of PSNR and 0.9944 of SSIM, even for images that are degraded by heavy rain.

Table 4. Comparison of the average values of PSNR (in dB) and SSIM of the reconstructed light, medium, and heavy rainy images.

Rain Level	PSNR (dB)	SSIM
Light	45.15	0.9987
Medium	43.36	0.9974
Heavy	40.10	0.9944

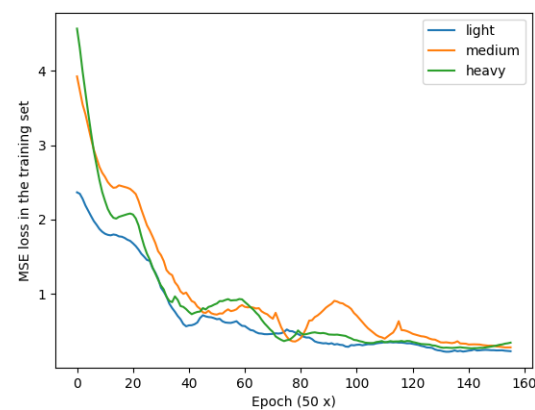


Figure 8. Curves of \mathcal{L}_{DR} on the training set for light, medium, and heavy rainy images.

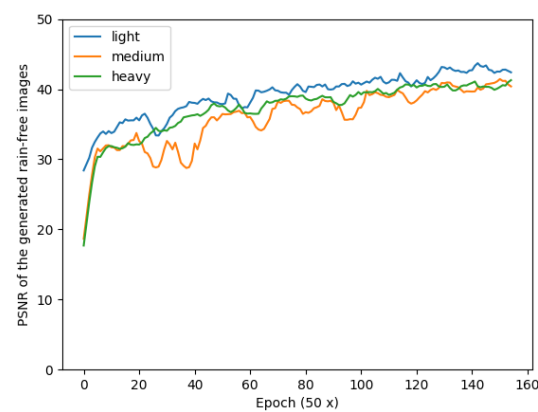


Figure 9. PSNR of the reconstructed rain-free images from light, medium, and heavy rainy images.

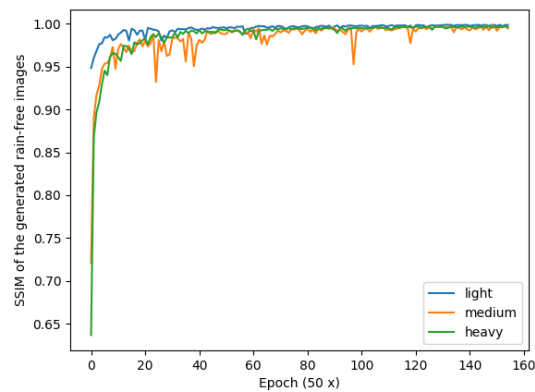


Figure 10. SSIM of the reconstructed rain-free images from light, medium, and heavy rainy images.

5.3.4. Visualization of Reconstructed Rain-Free Images

We present the visual comparison of images reconstructed by our Deraining Subnetwork below. Figure 11a–d correspond to the original rainy images, the patches cropped from the original ones, the reconstructed rain-free patches, and their corresponding ground truth (clean patches), respectively. Comparing patches in Figure 11b,c, we can observe that our Deraining Subnetwork can effectively remove rain streaks from the rainy input images. Comparing patches in Figure 11c,d, the reconstructed patches can preserve image details, e.g., the texture of cars.



Figure 11. Visualization of images and patches reconstructed by Deraining Subnetwork. (a) The original input images. (b) The patches cropped from the original image. (b) The reconstructed rain-free patches. (c) The corresponding ground truth.

Furthermore, we also used images degraded by medium-level rain as examples to visualize the reconstruction process of rain-free in Figure 12. Figure 12a,f are the input rainy patches and the corresponding ground truths. Figure 12b–e are the rain-free patches reconstructed by the proposed Deraining Subnetwork trained with 1, 50, 100, and 6000 epochs,

respectively. These visualization results demonstrate that the Deraining Subnetwork can remove rain streaks and restore the contents of input images.

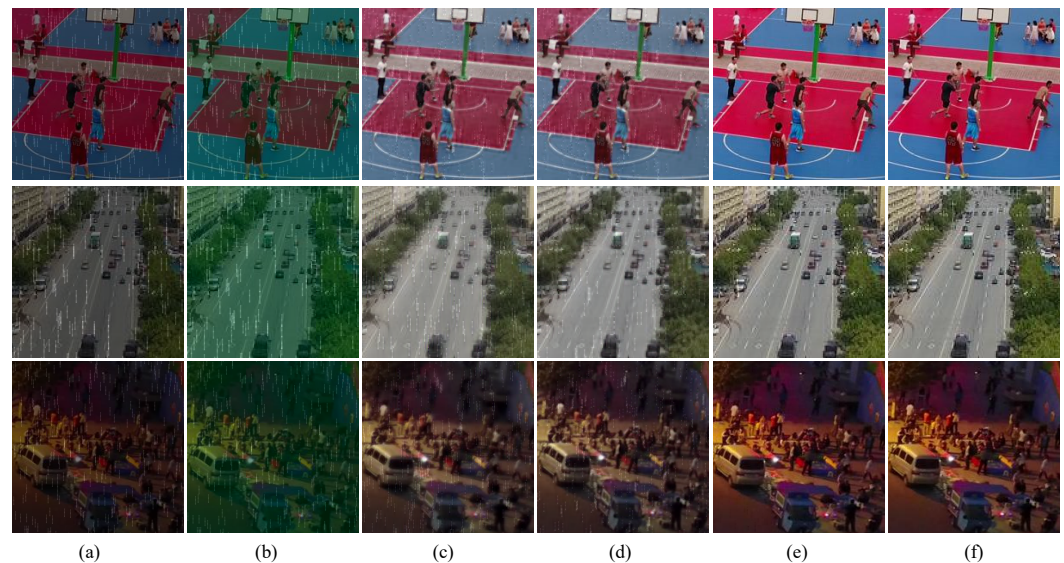


Figure 12. Visualization of the process of reconstructing rain-free patches with different epochs of training. Patches in Column (a) are the input rainy patches; patches in Columns (b–e) are the rain-free patches reconstructed by the proposed deraining model trained with 1, 50, 100, and 6000 epochs, respectively. Finally, patches in Column (f) are their corresponding ground truths.

5.4. Comparison with SOTA Methods

There is a tradition to present the SOTA comparison results in Table 2.

5.4.1. General Object Detectors

CoDerainNet is compared with two current SOTA object detectors, YOLOv5 and YOLOv7. Table 5 presents that CoDerainNet obtains the best detection performance on the three datasets. Specifically, on the RainVisDrone dataset, the AP_{50} obtained with CoDerainNet is 5.40% and 2.66% higher than those obtained with YOLOv5 and YOLOv7, respectively. On the RainUAVDT dataset, the AP_{50} of CoDerainNet is 4.77% and 2.08% higher than those obtained with YOLOv5 and YOLOv7, respectively. On the RainDrone dataset, the AP_{50} of CoDerainNet is 1.73% and 1.12% higher than those obtained by YOLOv5 and YOLOv7, respectively.

The two reasons for CoDerainNet's performance gain in accuracy are the reduction in noise in the features and the feature enhancement for small targets. CoDerainNet can reduce the noise in features caused by rain streaks with the Deraining Subnetwork under our ColTeaching paradigm and thus improves the features' quality for the task of DroneDet. Furthermore, our DroneDet Subnetwork integrates context collector and the fine-grained target-focusing module to enhance small targets' features.

5.4.2. ImDerain-Based and DA-Based Object Detectors

We compared the detection accuracy of CoDerainNet with those of four ImDerain-based detectors. The two deraining models, MPRNet and PReNet, are trained on the synthetic rainy datasets for image deraining. We remove rain streaks using the two well-trained deraining models to generate rain-free images from the synthetic and real rainy datasets. Then, the two detectors, YOLOv5 and YOLOv7, are trained using these generated images. In the end, we obtained the detection accuracy of the four concatenation models on the generated rain-free images.

Table 5 presents the detection accuracy of our CoDerainNet and the four combination models. Our CoDerainNet obtains AP_{50} values of 62.15%, 37.50%, and 59.21% on the three datasets, which are comparable with that of MPRNet-YOLOv7. However, the inference time of our CoDerainNet is less than that of MPRNet-YOLOv7. The comparison results

demonstrate that our CoDerainNet is more efficient in terms of inference speed and computational costs with comparable accuracy with the ImDerain-based detectors. In addition, Table 5 also shows the performance comparison of CoDerainNet with AdaptiveTeacher, and CoDerainNet outperforms AdaptiveTeacher.

ImDerain-based detectors adopt complex deraining models to improve DroneDet accuracy effectively. These deraining models gradually learn restoration functions in multi-stage architecture to achieve notable performance. As a result, these methods substantially increase the overall complexity of detectors. For example, the computational complexity of MPRNet-YOLOv5 is about 20 times higher than that of YOLOv5 (2408.40 G vs 108.40 G). However, our CoDerainNet can achieve a DroneDet performance comparable to the ImDerain-based detectors while only slightly increasing the computational costs, since the Deraining Subnetwork is only executed during training but not inference.

5.4.3. Visualization of Detection Results

We compared the visual detection results of a rainy image from RainVisDrone. Observations in Figure 13 indicate that CoDerainNet detects objects more accurately than the other models. In addition, Figures 14–16 visualize the detection results obtained by CoDerainNet on rainy images. We can observe that CoDerainNet obtains impressive detection results on both synthetic and real rainy datasets. It is worth mentioning that CoDerainNet can detect targets in low-light images. According to the confidence score, there is still much room for improvement in low-light detection performance.

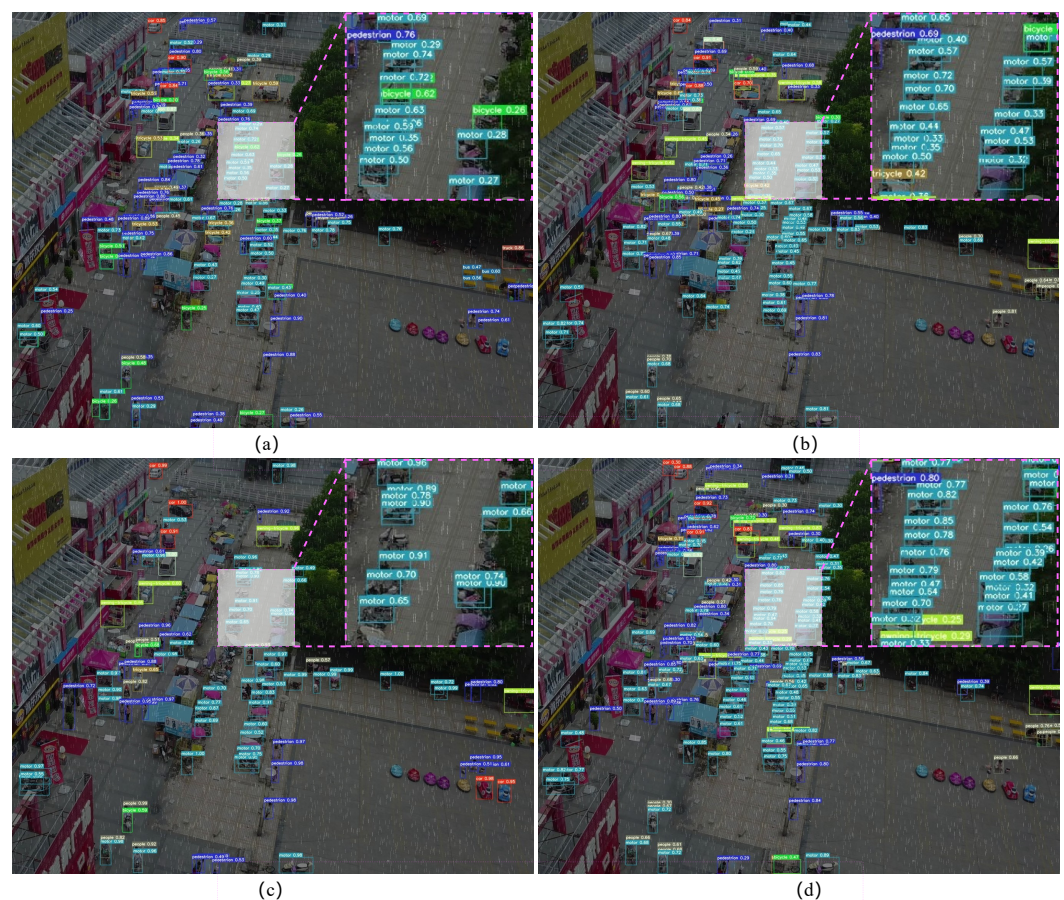
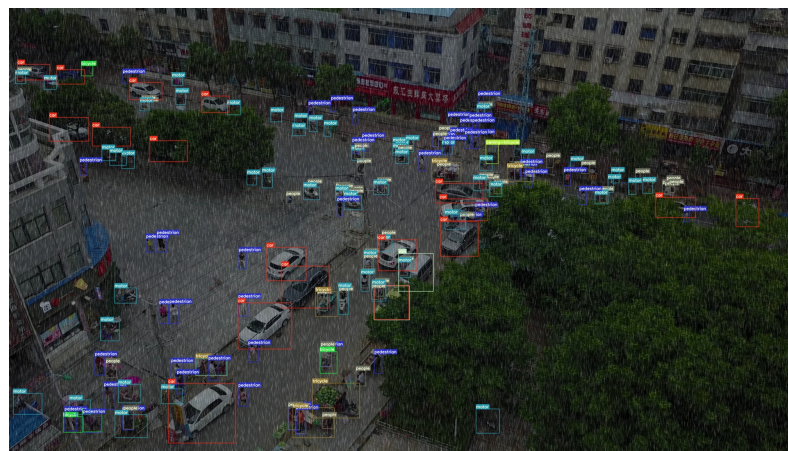


Figure 13. Comparison of the detection results of a medium rainy image taken from the RainVisDrone dataset. (a) YOLOv7, (b) PReNet-YOLOv7, (c) AdaptiveTeacher, and (d) CoDerainNet.

Table 5. Comparison of CoDerainNet with SOTA detectors on RainVisDrone, RainUAVDT, and RainDrone datasets.

Method	Reference	RainVisDrone			RainUAVDT			RainDrone		
		AP(%)	AP ₅₀ (%)	AP ₇₅ (%)	AP(%)	AP ₅₀ (%)	AP ₇₅ (%)	AP(%)	AP ₅₀ (%)	AP ₇₅ (%)
YOLOv5	Github 21	36.57	56.75	37.98	21.67	32.73	25.83	38.46	57.48	40.23
YOLOv7	arXiv 22	36.49	59.49	37.80	21.14	35.42	23.65	39.31	58.09	41.86
PReNet-YOLOv5	CVPR 19	37.22	58.72	38.96	23.29	34.58	28.19	38.89	57.96	40.72
MPRNet-YOLOv5	CVPR 21	37.24	59.59	39.05	24.50	36.41	29.18	39.76	58.18	41.03
PReNet-YOLOv7	CVPR 19	38.64	62.06	40.43	24.05	39.27	27.49	40.21	59.81	42.41
MPRNet-YOLOv7	CVPR 21	39.59	63.01	41.47	25.01	40.08	28.91	40.74	59.90	42.79
AdaptiveTeacher	CVPR 22	18.03	31.80	18.13	14.34	25.43	15.41	19.67	30.98	19.03
CoDerainNet	Ours	38.75	62.15	40.77	23.82	37.50	28.21	39.98	59.21	40.89

**Figure 14.** Visualization of detection results by CoDerainNet on a heavy rainy image in RainVisDrone.**Figure 15.** Visualization of detection results by applying CoDerainNet to a low-light rainy image in RainVisDrone.

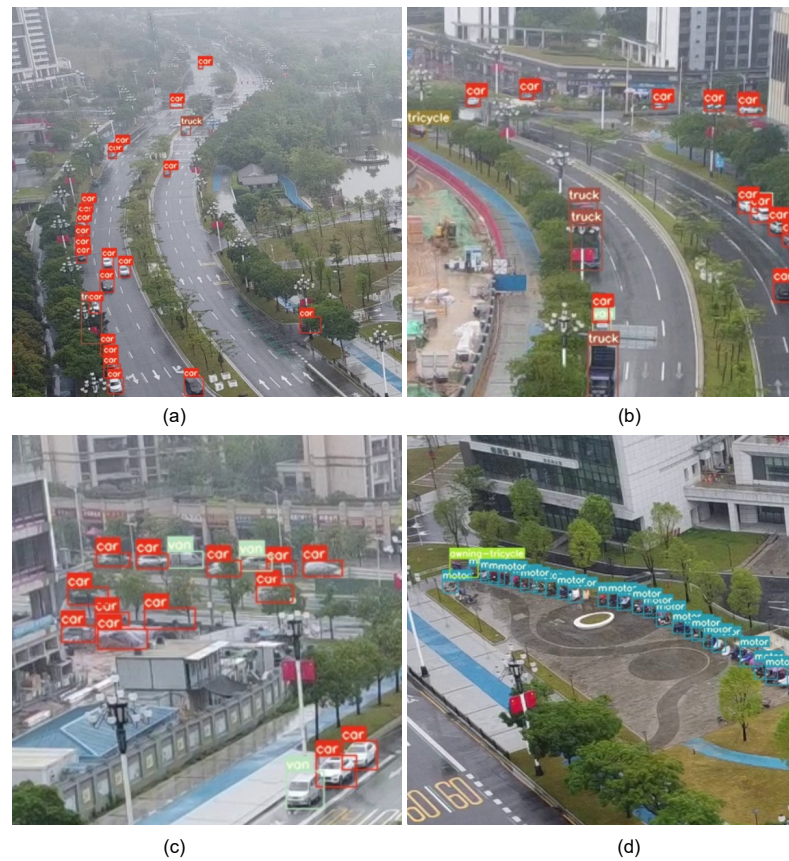


Figure 16. Visualization of the detection results obtained with CoDerainNet on our RainDrone dataset. These subfigures (a), (b), (c), and (d) are captured by drones in real rainy weather conditions.

5.4.4. Overall Complexity Comparison

We compared the overall complexity of CoDerainNet with that of SOTA methods. The evaluation metrics included FLOPs (Floating Point Operations, the calculation amount of a model) and processing time (milliseconds) on the server. Table 6 shows that CoDerainNet recorded a processing time of 106.78 s and FLOPs of 241.59. Compared with the ImDerain-based methods, including PReNet-YOLOv5, MPRNet-YOLOv5, PReNet-YOLOv7, and MPRNet-YOLOv7, CoDerainNet significantly reduces the inference time and calculation amount because our approach does not need image deraining as a pre-processing step, which involves a high computation cost. Compared with the general object detectors, including YOLOv5 and YOLOv7, our CoDerainNet slightly increased the inference time and calculation amount because the context collector and the fine-grained target-focusing module are integrated into our DroneDet Subnetwork to boost the performance of small object detection.

Table 6. Comparison of different methods in terms of inference time and calculation amount.

Method	Time (Milliseconds)	FLOPs (G)
YOLOv5	55.10	108.40
YOLOv7	27.89	103.30
PReNet-YOLOv5	386.15	1682.35
MPRNet-YOLOv5	270.10	2408.40
PReNet-YOLOv7	358.94	1677.25
MPRNet-YOLOv7	242.89	2403.30
AdaptiveTeacher	119.17	258.98
CoDerainNet	106.78	241.59

6. Limitation

Although our proposed CoDerainNet obtained promising results on both synthetic and real rainy datasets, the performance improvement on the real dataset is not as significant as that on the synthetic datasets. An observation in Table 5 presents that a gain (1.73%) of AP_{50} obtained by CoDerainNet on the real dataset RainDrone is less than the gain (5.4%) of AP_{50} on the synthetic dataset RainVisdrone. The main reason for this phenomenon is that only synthetic training pairs of clean and rainy images are used for training the Deraining Subnetwork. Real training ones are often unavailable. There is a big difference between synthetic and real rain streaks in terms of direction and density. Therefore, the lack of training image pairs under real rainy weather is the main cause of the poor performance in the real rainy dataset. Semi-supervised learning for image deraining may have the potential to address this limitation.

7. Conclusions

We proposed CoDerainNet to improve Rainy DroneDet with slightly increased computational costs. CoDerainNet is an interactively trained Deraining Subnetwork and DroneDet Subnetwork through our ColTeaching paradigm. Our key idea was to transfer the intrinsic degradation knowledge from the Deraining Subnetwork to the DroneDet Subnetwork and teaches the DroneDet Subnetwork such knowledge to suppress the impact of rain-specific interference on features extracted for DroneDet. Three new drone-captured datasets, i.e., RainVisDrone, RainUVADT, and RainDrone, were also built for interactive detection and deraining. Extensive experiments demonstrated that CoDerainNet can obtain better detection results. The results also demonstrated its effectiveness in a real rainy scenario.

In the near future, we can extend our CoDerainNet in the following directions. Firstly, we can adapt it to other kinds of challenging weather conditions, including foggy weather, snowy weather, and nighttime conditions. Secondly, we can investigate a simple semi-supervised learning framework for deraining images collected in real rainy scenarios.

Author Contributions: Y.X. conceived of and designed the experiments and wrote the paper; Y.X. performed the experiments and original draft preparation; Y.X. and W.J. analyzed the data; W.J., J.F., X.L. and F.L. provided the review and editing; Q.M. supervised this study and reviewed the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Not applicable.

Acknowledgments: We sincerely thank the authors of Yolov5 for releasing the source codes.

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

References

1. Deng, S.; Li, S.; Xie, K.; Song, W.; Liao, X.; Hao, A.; Qin, H. A global-local self-adaptive network for drone-view object detection. *IEEE Trans. Image Process.* **2020**, *30*, 1556–1569. [[CrossRef](#)] [[PubMed](#)]
2. Huang, Y.; Chen, J.; Huang, D. UFPMP-Det: Toward accurate and efficient object detection on drone imagery. In Proceedings of the AAAI Conference on Artificial Intelligence, Vancouver, BC, Canada, 28 February–1 March 2022; Volume 36, pp. 1026–1033.
3. Chalavadi, V.; Jeripothula, P.; Datla, R.; Ch, S.B. mSODANet: A Network for Multi-Scale Object Detection in Aerial Images using Hierarchical Dilated Convolutions. *Pattern Recognit.* **2022**, *126*, 108548. [[CrossRef](#)]
4. Yang, F.; Fan, H.; Chu, P.; Blasch, E.; Ling, H. Clustered object detection in aerial images. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 8311–8320.
5. Yang, W.; Tan, R.T.; Feng, J.; Liu, J.; Guo, Z.; Yan, S. Deep joint rain detection and removal from a single image. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1357–1366.
6. Li, Y.; Tan, R.T.; Guo, X.; Lu, J.; Brown, M.S. Rain streak removal using layer priors. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 2736–2744.
7. Qian, R.; Tan, R.T.; Yang, W.; Su, J.; Liu, J. Attentive generative adversarial network for raindrop removal from a single image. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 2482–2491.

8. Ren, D.; Zuo, W.; Hu, Q.; Zhu, P.; Meng, D. Progressive image deraining networks: A better and simpler baseline. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 3937–3946.
9. Araujo, I.B.; Tokuda, E.K.; Cesar, R.M. The Impact of Real Rain in a Vision Task. In *Proceedings of the ECCV*; Springer: Berlin, Germany, 2020; pp. 291–298.
10. Zamir, S.W.; Arora, A.; Khan, S.; Hayat, M.; Khan, F.S.; Yang, M.H.; Shao, L. Multi-stage progressive image restoration. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 14821–14831.
11. Wei, W.; Meng, D.; Zhao, Q.; Xu, Z.; Wu, Y. Semi-supervised transfer learning for image rain removal. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 3877–3886.
12. Vs, V.; Gupta, V.; Oza, P.; Sindagi, V.A.; Patel, V.M. Mega-cda: Memory guided attention for category-aware unsupervised domain adaptive object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 4516–4526.
13. Sindagi, V.A.; Oza, P.; Yasarla, R.; Patel, V.M. Prior-based domain adaptive object detection for hazy and rainy conditions. In *Proceedings of the ECCV*; Springer: Berlin, Germany, 2020; pp. 763–780.
14. Lee, S.; Son, T.; Kwak, S. Fifo: Learning fog-invariant features for foggy scene segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 18911–18921.
15. Zhu, P.; Wen, L.; Du, D.; Bian, X.; Fan, H.; Hu, Q.; Ling, H. Detection and Tracking Meet Drones Challenge. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *44*, 7380–7399. [[CrossRef](#)] [[PubMed](#)]
16. Du, D.; Qi, Y.; Yu, H.; Yang, Y.; Duan, K.; Li, G.; Zhang, W.; Huang, Q.; Tian, Q. The unmanned aerial vehicle benchmark: Object detection and tracking. In *Proceedings of the ECCV*; Springer: Berlin, Germany, 2018; pp. 370–386.
17. Li, B.; Peng, X.; Wang, Z.; Xu, J.; Feng, D. Aod-net: All-in-one dehazing network. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 4770–4778.
18. Qin, X.; Wang, Z.; Bai, Y.; Xie, X.; Jia, H. FFA-Net: Feature fusion attention network for single image dehazing. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 11908–11915.
19. Liu, W.; Ren, G.; Yu, R.; Guo, S.; Zhu, J.; Zhang, L. Image-Adaptive YOLO for Object Detection in Adverse Weather Conditions. In Proceedings of the AAAI Conference on Artificial Intelligence, Washington, DC, USA, 22 February–1 March 2022; Volume 36, pp. 1792–1800.
20. He, Z.; Zhang, L. Multi-adversarial faster-rcnn for unrestricted object detection. In Proceedings of the Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Korea, 27 October–2 November 2019; pp. 6668–6677.
21. Xi, Y.; Jia, W.; Zheng, J.; Fan, X.; Xie, Y.; Ren, J.; He, X. DRL-GAN: dual-stream representation learning GAN for low-resolution image classification in UAV applications. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2020**, *14*, 1705–1716. [[CrossRef](#)]
22. Qiu, H.; Li, H.; Wu, Q.; Meng, F.; Xu, L.; Ngan, K.N.; Shi, H. Hierarchical context features embedding for object detection. *IEEE Trans. Multimed.* **2020**, *22*, 3039–3050. [[CrossRef](#)]
23. Li, G.; Liu, Z.; Zeng, D.; Lin, W.; Ling, H. Adjacent Context Coordination Network for Salient Object Detection in Optical Remote Sensing Images. *IEEE Trans. Cybern.* **2022**, 1–13. [[CrossRef](#)] [[PubMed](#)]
24. Tan, M.; Pang, R.; Le, Q.V. Efficientdet: Scalable and efficient object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 10781–10790.
25. Li, J.; Zhu, S.; Gao, Y.; Zhang, G.; Xu, Y. Change Detection for High-Resolution Remote Sensing Images Based on a Multi-Scale Attention Siamese Network. *Remote Sens.* **2022**, *14*, 3464. [[CrossRef](#)]
26. Xi, Y.; Jia, W.; Miao, Q.; Liu, X.; Fan, X.; Li, H. FiFoNet: Fine-Grained Target Focusing Network for Object Detection in UAV Images. *Remote Sens.* **2022**, *14*, 3919. [[CrossRef](#)]
27. Tian, J.; Wang, B.; Guo, R.; Wang, Z.; Cao, K.; Wang, X. Adversarial Attacks and Defenses for Deep-Learning-Based Unmanned Aerial Vehicles. *IEEE Internet Things J.* **2022**, *9*, 22399–22409. [[CrossRef](#)]
28. Ko, Y.; Kim, J.; Duguma, D.G.; Astillo, P.V.; You, I.; Pau, G. Drone Secure Communication Protocol for Future Sensitive Applications in Military Zone. *Sensors* **2021**, *21*, 2057. [[CrossRef](#)] [[PubMed](#)]
29. Krichen, M.; Adoni, W.Y.H.; Mihoub, A.; Alzahrani, M.Y.; Nahhal, T. Security Challenges for Drone Communications: Possible Threats, Attacks and Countermeasures. In Proceedings of the 2022 2nd International Conference of Smart Systems and Emerging Technologies (SMARTTECH), Riyadh, Saudi Arabia, 22–24 May 2022; pp. 184–189.
30. Zhang, Y.; Yang, Q. A Survey on Multi-Task Learning. *IEEE Trans. Knowl. Data Eng.* **2022**, *34*, 5586–5609. [[CrossRef](#)]
31. Zhao, H.; Shi, J.; Qi, X.; Wang, X.; Jia, J. Pyramid scene parsing network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2881–2890.
32. 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.
33. Rezatofighi, H.; Tsoi, N.; Gwak, J.; Sadeghian, A.; Reid, I.; Savarese, S. Generalized intersection over union: A metric and a loss for bounding box regression. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 658–666.
34. Zhang, H.; Patel, V.M. Density-aware single image de-raining using a multi-stream dense network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 695–704.

35. Huang, S.C.; Hoang, Q.V.; Le, T.H. SFA-Net: A Selective Features Absorption Network for Object Detection in Rainy Weather Conditions. *IEEE Trans. Neural Networks Learn. Syst.* **2022**, 1–11. [[CrossRef](#)] [[PubMed](#)]
36. Jocher, G. YOLOv5 Source Code. Available online: <https://github.com/ultralytics/yolov5> (accessed on 1 August 2022).
37. Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; Zagoruyko, S. End-to-end object detection with transformers. In *Proceedings of the ECCV*; Springer: Berlin, Germany, 2020; pp. 213–229.
38. Li, Y.J.; Dai, X.; Ma, C.Y.; Liu, Y.C.; Chen, K.; Wu, B.; He, Z.; Kitani, K.; Vajda, P. Cross-Domain Adaptive Teacher for Object Detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022*; pp. 7581–7590.
39. Zhu, X.; Lyu, S.; Wang, X.; Zhao, Q. TPH-YOLOv5: Improved YOLOv5 Based on Transformer Prediction Head for Object Detection on Drone-Captured Scenarios. In *Proceedings of the ICCV Workshops, Montreal, BC, Canada, 11–17 October 2021*; pp. 2778–2788.
40. Everingham, M.; Eslami, S.M.A.; Van Gool, L.; Williams, C.K.I.; Winn, J.; Zisserman, A. The Pascal Visual Object Classes Challenge: A Retrospective. *Int. J. Comput. Vis.* **2015**, *111*, 98–136. [[CrossRef](#)]
41. Zhang, K.; Zuo, W.; Chen, Y.; Meng, D.; Zhang, L. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE Trans. Image Process.* **2017**, *26*, 3142–3155. [[CrossRef](#)] [[PubMed](#)]
42. Wang, Z.; Bovik, A.C.; Sheikh, H.R.; Simoncelli, E.P. Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Image Process.* **2004**, *13*, 600–612. [[CrossRef](#)] [[PubMed](#)]
43. Huynh-Thu, Q.; Ghanbari, M. Scope of validity of PSNR in image/video quality assessment. *Electron. Lett.* **2008**, *44*, 800–801. [[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.