drones

MDPI

Article

GGT-YOLO: A Novel Object Detection Algorithm for Drone-Based Maritime Cruising

Yongshuai Li [1], Haiwen Yuan [1,2,3,*], Yanfeng Wang [4] and Changshi Xiao [3,4]

[1] School of Electrical and Information Engineering, Wuhan Institute of Technology, Wuhan 430205, China
[2] Intelligent Transportation Systems Research Center, Wuhan University of Technology, Wuhan 430063, China
[3] National Engineering Research Center for Water Transport Safety, Wuhan University of Technology, Wuhan 430063, China
[4] School of Navigation, Wuhan University of Technology, Wuhan 430063, China
* Correspondence: hw_yuan@whut.edu.cn

Abstract: Drones play an important role in the development of remote sensing and intelligent surveillance. Due to limited onboard computational resources, drone-based object detection still faces challenges in actual applications. By studying the balance between detection accuracy and computational cost, we propose a novel object detection algorithm for drone cruising in large-scale maritime scenarios. Transformer is introduced to enhance the feature extraction part and is beneficial to small or occluded object detection. Meanwhile, the computational cost of the algorithm is reduced by replacing the convolution operations with simpler linear transformations. To illustrate the performance of the algorithm, a specialized dataset composed of thousands of images collected by drones in maritime scenarios is given, and quantitative and comparative experiments are conducted. By comparison with other derivatives, the detection precision of the algorithm is increased by 1.4%, the recall is increased by 2.6% and the average precision is increased by 1.9%, while the parameters and floating-point operations are reduced by 11.6% and 7.3%, respectively. These improvements are thought to contribute to the application of drones in maritime and other remote sensing fields.

Keywords: drone; maritime surveillance; object detection; Transformer; GhostNet

Citation: Li, Y.; Yuan, H.; Wang, Y.; Xiao, C. GGT-YOLO: A Novel Object Detection Algorithm for Drone-Based Maritime Cruising. *Drones* **2022**, *6*, 335. https://doi.org/10.3390/drones6110335

Academic Editor: Anastasios Dimou

Received: 3 October 2022
Accepted: 28 October 2022
Published: 31 October 2022

1. Introduction

The global market of drones is expected to exceed $48 billion by 2026, which has been reported by Drone Industry Insights [1]. Given their advantages of high mobility, rapid response and great view, drones are playing an important role in various human social activities, e.g., monitoring [2,3], photogrammetry [4,5], search-and-rescue [6], etc. Advanced Artificial Intelligence and Internet of Things techniques have been equipped with drones to carry out these tasks autonomously. However, there exist challenges to be addressed in real-world applications.

Object detection helps drones to find the position and class of objects in their view and is the primary requirement for drones applied in maritime cruising and searching missions. For the last twenty years, various algorithms and application scenarios have been studied for object detection. For traditional approaches, handcrafted features are extracted from the patches of images and one or multiple classifiers are selected to traverse the total image, e.g., histogram of oriented gradient (HOG) detector, deformable parts model (DPM), etc. [7]. As popular solutions in the last ten years, deep-learning-based approaches utilize deep network to learn high-level feature representations of various objects, e.g., region convolutional neural network (R-CNN), you only look once series (YOLOs), etc. [8]. Even though there have been remarkable achievements using the above approaches, some common challenges remain to be addressed, such as object rotation and scale changes, small and occluded object detection, real-time of onboard system, etc. For traditional scenarios, pedestrian and vehicles as main detected objects present relatively

stable and consistent appearance in the view of the drone, e.g., VisDrone Dataset [9], Okutama-Action Dataset [10], etc. Moreover, the variations of illumination and orientation are more distinct in large-scale, dynamic maritime scenarios. Affected by the above factors, ships are displayed as small or occluded objects in a moving perspective, which might seriously reduce the performance of object detection algorithms. At the same time, high computational efficiency is required by the onboard systems of drones.

Either detection accuracy or computational cost are a serious requirement for drone-based maritime object detection. YOLOv5 [11] presents four versions for different application scenarios, and YOLOv5s (as one of the versions) displays appropriate performance for detection accuracy and computational cost, which make it possible to achieve object detection on board drones. Hence, YOLOv5 is selected and studied for drone-based maritime object detection. Using YOLOv5 as the framework, Transformer [12] is fused with the backbone network to enhance feature extraction, which is beneficial for detecting small or occluded objects in maritime scenarios. Meanwhile, GhostNet [13] is utilized to replace the ordinary convolution in the network with linear transformations, which require fewer parameters and lesser computation cost. To evaluate the proposed algorithm, real image or video data have been collected during maritime cruise missions using drones. Quantitative and comparative experiments are conducted, and the results are analyzed at the end of the paper. The acquired conclusion reveals the advantages of the algorithm in real-world maritime scenarios.

The remainder of this paper is laid out as follows: Section 2 describes related works on drone-based object detection and the applications. The specialized dataset and drone-based maritime object detection algorithm are presented in Section 3. Details of the experiments and analysis are given in Section 4. In Section 5 we draw a conclusion and the direction for future research is also identified.

2. Related Work

In recent years, object detection based on drone vision has been studied extensively for various application fields. Related works are selected to introduce in this section.

With the advantages of great perspective and high resolution, drone vision is very suitable for remote sensing. Travelling vehicles [14], road information [15] and pavement distress [16] could be extracted from drone imagery by deep learning algorithms, e.g., Faster R-CNN, YOLOs, etc. An improved Faster R-CNN consisting of a top-down-top feature pyramid fusion structure is proposed for visual detection tasks of catenary support devices defect [17]. For small object detection in drone images, more abundant feature information could be extracted by a multi-branch parallel feature pyramid network [18]. Furthermore, a supervised spatial attention mechanism was considered to reduce the background noise. Small object detection accuracy could be improved by feature pyramid network, which is capable of fusing more representative features including shallow and deep feature maps [19]. The receptive field for small object detection was enriched by concatenating two ResNet models in the DarkNet of YOLOv3 [20] and increasing convolution operations in an early layer [21]. To minimize the occurrence of missed targets due to occlusion, Tan et al. [22] introduced soft non-maximum suppression into the framework of YOLOv4 [23]. YOLOv5 presents four versions for different application scenarios: YOLOv5s, YOLOv5m, YOLOv5l and YOLOv5x. Small object detection by the vision of drones has been studied by improving YOLOv5 [24]. The refinements, including adding a microscale detection layer, setting prior anchor boxes and adapting the confidence loss function of the detection layer, were implemented in the YOLOv5 framework for small-sized wheat spike detection [25]. Thus, it can be seen that multiscale representation, contextual information, super resolution, and region proposal are the main solutions to improve the performance of small object detection [26]. YOLOv6 [27] and YOLOv7 [28] have been proposed successively in 2022. The backbone of YOLOv6 utilizes EfficientRep instead of CSPDarkNet. It is worth mentioning that YOLOv6 continues to use anchor-free. A new border regression loss SIOU is introduced; in other words, YOLOv6 is the best combination of YOLOv5 and

YOLOx. YOLOv7 presents a planned re-parameterized model to replace some original modules. Due to time limitations, the related work on their applications is rare.

Considering the limited onboard computation resource, a few lightweight networks have been proposed for drone vision. To reduce the computational cost and network size, pointwise convolution and regular convolution were combined as the main building block of the network proposed by Liu et al. [29]. The inverted residual block [30] was utilized to construct a lightweight network for object recognition. For vehicle detection in aerial images, Javadi et al. [31] optimized YOLOv3 by replacing Darknet-53 with MobileNet-v2 which integrates deep separable convolution, linear bottleneck and inverted residual. An improved network named MobileNet-v3 was realized by adding lightweight attention model and h-swipe into MobileNet-v2. MobileNet-v3 was used for reducing the computation cost of YOLOv4 while ensuring feature extraction from the aerial images [32]. Thus, how to obtain a good trade-off between computational cost and detection accuracy has become the focus of drone-vision research [33].

Maritime object detection, as one typical scenario, has been studied for many years. Prasad et al. [34,35] summarized the visual perception algorithms for maritime scenarios in recent years and proposed the corresponding assessment criteria of maritime computer vision. The maritime datasets were provided for training and evaluating the deep-learning-based visual algorithms in [36,37]. The multi-spectral vision was studied for human body detection in the maritime search-and-rescue tasks using drones [38]. Reverse depthwise separable convolution was applied in the backbone model of YOLOv4 [39], which reduced the network parameters by 40% and was suitable for vision-based surface target detection of unmanned ships. Ship re-identification [40] is significant when ships frequently move in and out of the drone's view. Even though various algorithms for ship detection in SAR or horizontal perspective images were presented in [41–43], drone vision-based maritime object detection still presents some challenges. Background variation, scale variation, illumination conditions, visible proportion, etc. are thought to be especially serious while detecting and tracking maritime objects using drone vision.

Inspired by the outstanding works mentioned above, both detection accuracy and computation cost are required by mobile vision. As one typical one-stage object detection framework, YOLO series have been studied and applied in drone-based vision systems. On the one hand, feature enhancement is the main way to improve detection accuracy. On the other hand, network lightweight is thought to reduce the computational burden of onboard systems. As a result, YOLOv5 is studied as the framework of our drone vision for maritime object detection in this work. To achieve our aim, advanced models such as Transformer and GhostNet are utilized to improve the accuracy and efficiency of the original YOLOv5. Comparative experiments are conducted to obtain the optimal solution regarding re-configuring the YOLOv5 with Transformer and GhostNet. The improved object detection framework is expected to have better performance in drone maritime cruise scenarios.

3. Materials and Methods

Maritime object detection based on drone vision is studied in this work. While drones are cruising through typical maritime scenarios, various appearances of ships of different scales would be presented in their view. Therefore, detection accuracy and computation efficiency of the algorithm have to be considered for detecting these maritime objects. To achieve these, a novel drone-based maritime object detection algorithm is presented in Figure 1. The algorithm can be mainly divided into three parts: The backbone is responsible for extracting features from an input image and is composed of three network layers based on CNN and Transformer. The feature maps with scales of 80×80, 40×40 and 20×20 can be calculated through the backbone. The neck is responsible for fusing the feature maps. For the head, three detectors at different scales are utilized to calculate the positions and sizes of objects. In addition, the dataset specialized for drone-based maritime object detection is described in this section.

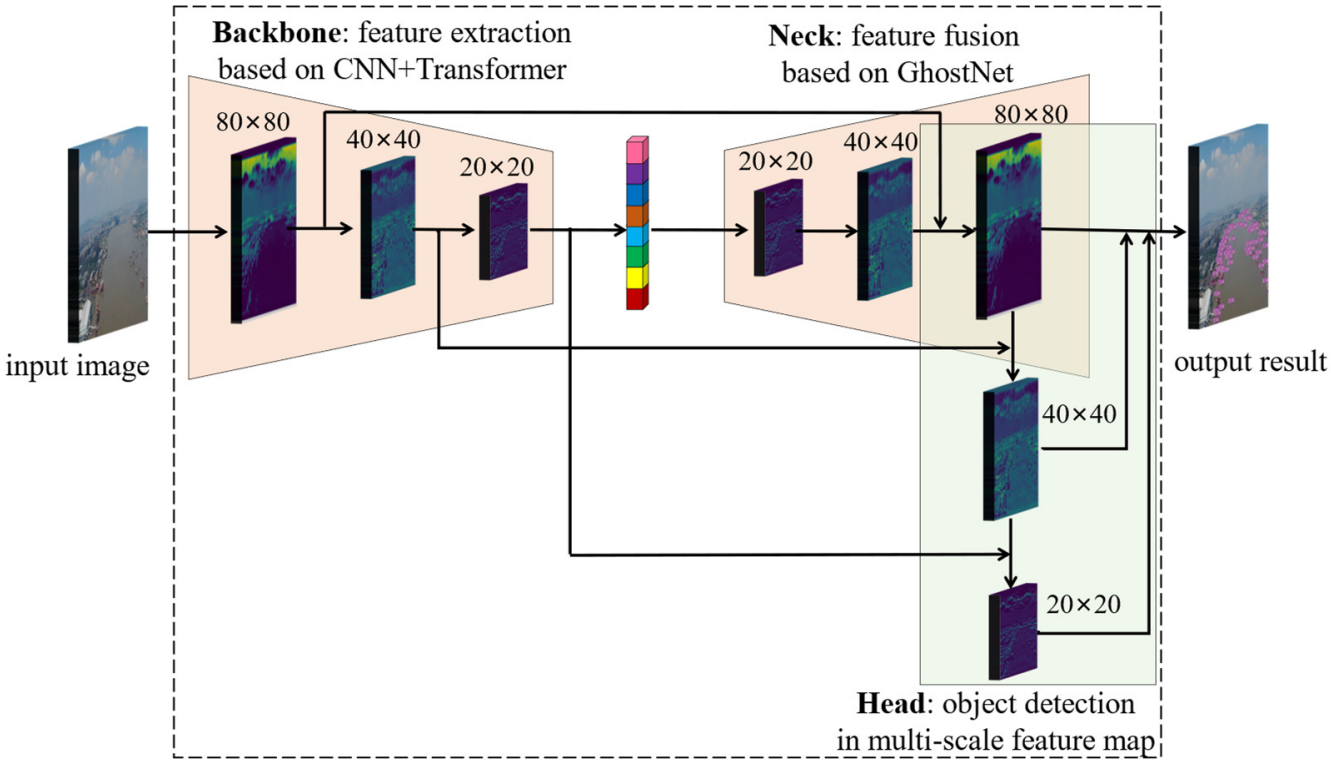


Figure 1. The framework of drone-based maritime object detection.

3.1. MariDrone Dataset

A specialized dataset is constructed for drone-based maritime object detection. The dataset is composed of thousands of maritime scenario images collected by our drone DJI M300, therefore named MariDrone. The drone with a size 810 × 670 × 430 mm has a payload capacity of 2.7 kg and the effective range of the remote control is 8 km. The positioning precision is 1~2 cm. The onboard vision system is deployed with a wide-angle camera and an embedded GPU device. The camera has a high resolution of 1200 million and the angle view is 82.9 deg. The embedded device based on the NVIDIA Pascal™ GPU architecture is equipped with 8 GB of memory and has a memory bandwidth of 59.7 GB/s. It is responsible for real time object detection using our algorithm. Images are collected by the onboard vision system when the drone is cruising over the Yangtze River. Regarding illumination, both sunny and cloudy conditions are involved in the dataset. The 3840 × 2140 image resolution is enough high to retain small objects or local details. In order to ensure the generalization of the MariDrone dataset, maritime videos are recoded using drones in different weather and illumination conditions. Through sampling these videos, a total of 4743 real images were obtained. Compared with other similar datasets, the MariDrone dataset was constructed completely by the flying drone. As a result, different scales, varying illuminations and various views are well presented in our dataset.

Furthermore, data augmentation is thought to extend the MariDrone dataset. As shown in Figure 2, random combinations of transformation operations involving translation, scaling, rotation, dithering, etc. are utilized in the data augmentation process. Translation, scaling and rotation can increase the forms of the labeled objects in the images. Meanwhile, maritime scenarios have been enriched by color dithering. Through such data augmentation, a total of 8340 images were composed for the MariDrone dataset. Each image was annotated according to COCO format. The dataset was divided into training-set, validation-set and test-set at a ratio of 7:2:1.

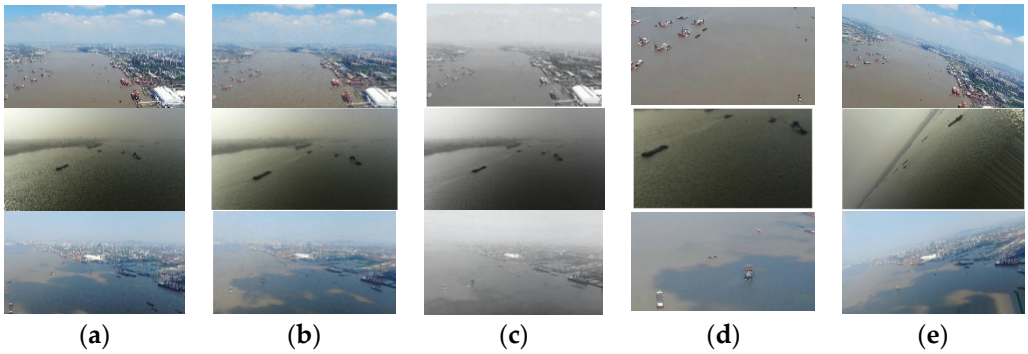


Figure 2. Data augmentation of MariDrone dataset. (**a**) Original image; (**b**) noise; (**c**) illumination; (**d**) scaling; (**e**) rotation.

3.2. GGT-YOLO Algorithm

The drone-based maritime object detection algorithm is described in this section, as shown in Figure 3. Using YOLOv5 as the framework, one Transformer is fused in the backbone to enhance the ability of feature extraction; it is of benefit to detect small or occluded objects from complex maritime scenarios in the view of drone. Two GhostNets are utilized to reduce the computational consumption of the network. Therefore, the algorithm is named GGT-YOLO. Compared with YOLOv5 and other derivatives, GGT-YOLO can achieve an optimal balance between detection accuracy and computational cost.

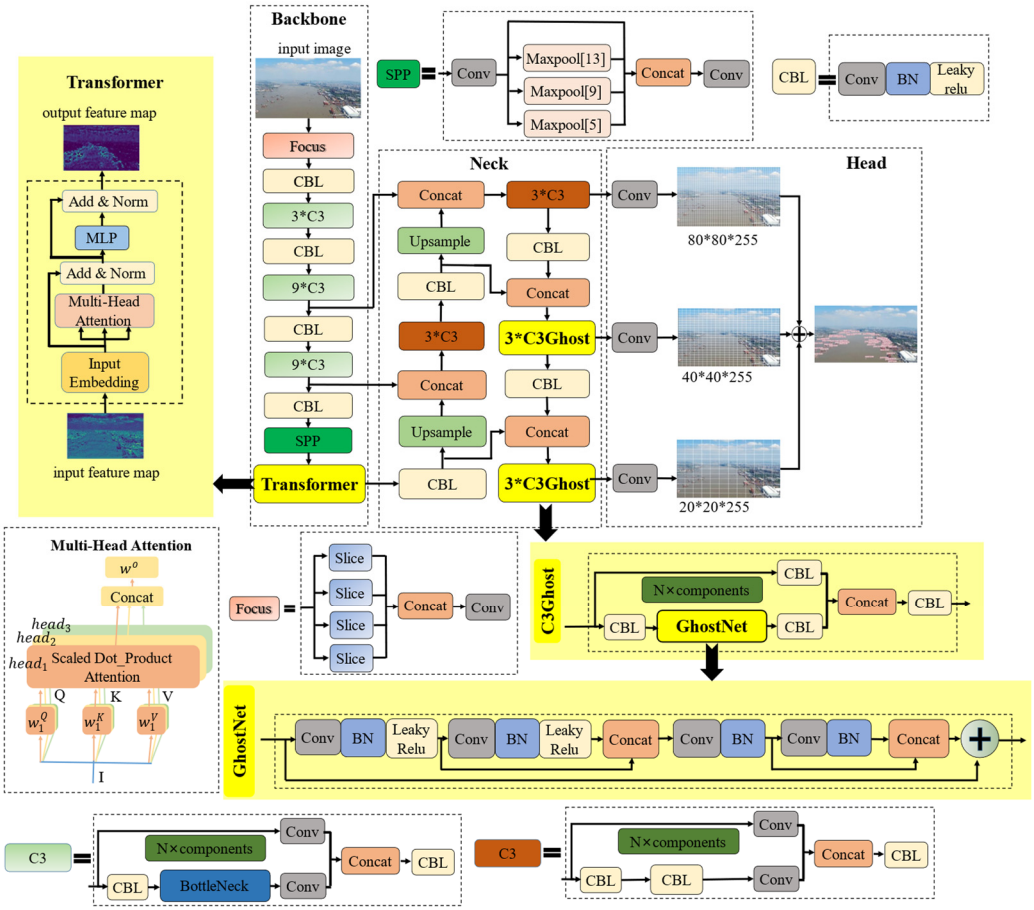


Figure 3. The details of GGT-YOLO architecture.

3.2.1. Object Detection Framework

Compared with the previous versions, YOLOv5 has advantages in data enhancement, feature extraction and loss calculation. Depending on faster detection speed and fewer computational requirements, YOLOv5s is a light version of YOLOv5 and is convenient for deploying onboard drones and other mobile terminals. Therefore, YOLOv5s is employed as the object detection framework in our work. Please note that in the following work any mention of 'YOLOv5' refers to YOLOv5s. Main sections of the framework are described: In the input section, input images are pre-processed to be standardized through mosaic data enhancement, anchor box calculation and image scaling orderly. Then, a backbone is established for extracting various features from standard images. In this section, a focus model is applied to calculate the reduced parameters through a series of slicing operations. CBL is defined as a specialized network involving convolution and batch normalization with activation function and used for transmitting features to alleviate the gradient vanishing. A cross-stage parallel network named C3 can expand the gradient path so as to enhance feature extraction. More fine-grained feature maps are acquired via concatenating CBL with C3. One such combination of CBL and C3 is applied repeatedly in the backbone network to calculate the feature maps with different scales. Spatial pyramid pooling (SPP) is used to reduce the feature loss due to image scaling and distorting. Subsequently, a neck section is mostly responsible for fusing the feature maps with different scales. Using a feature pyramid framework, a bottom-up path aggregation network is designed. C3 in the neck section is different from that in the backbone section. It plays a role of down-sampling operations during the fusion. Meanwhile, Concat refers to concatenating the feature maps after sampling. In the end, three detection heads composed of convolution operations are used to output the detection results with different scales. In the component of each head, one 3×3 convolution is responsible for feature integration, while one 1×1 convolution is used to adjust the number of channels. In the framework, detecting the objects of large, medium and small sizes can be carried out by calculating the feature maps with the scales of 80×80, 40×40 and 20×20, respectively.

Although the YOLOv5 displays good performance, there are still some challenges to be solved, especially when deployed on board light and flexible drones. To improve the detection performance on scale variation and computational cost, a novel algorithm GGT-YOLO is proposed by modifying the primary YOLOv5.

3.2.2. Feature Extraction Optimization

Due to the scale variations and frequent occlusions of ships displayed in the view of drone, it is a challenge for YOLOv5 to detect maritime objects. As a typical attention mechanism model, Transformer can pay more attention to key features instead of background or blank areas and thus is introduced to enhance the feature extraction of the algorithm. Inspired by Vision Transformer, Transformer is applied in the backbone of the GGT-YOLO, as shown in Figure 3.

Transformer is composed of a multi-head attention and a multilayer perceptron (MLP). Both residual connection (Add) and normalization (Norm) are applied between these networks. Multi-head attention can calculate the relationship among pixels in different positions to enhance the key features, especially for objects from multi-subspaces. In fact, each head of self-attention can be viewed as a subspace of information. As shown in Figure 3, feature maps from the backbone network will be reshaped to form a vector $\boldsymbol{I}$ by flattening operation. And the query vector $\boldsymbol{Q}$, the key vector $\boldsymbol{K}$ and the value vector $\boldsymbol{V}$ can be calculated from $\boldsymbol{I}$ by different linear transformations. Specifically, $head_i$ denotes the result of the i-th self-attention obtained by scaled dot-product attention, which is given as:

$$head_i = Attention(\boldsymbol{Q}_i, \boldsymbol{K}_i, \boldsymbol{V}_i) = Attention(\boldsymbol{I}\boldsymbol{W}_i^{\boldsymbol{Q}}, \boldsymbol{I}\boldsymbol{W}_i^{\boldsymbol{K}}, \boldsymbol{I}\boldsymbol{W}_i^{\boldsymbol{V}}), \quad (1)$$

where $\boldsymbol{IW}_i^{\boldsymbol{Q}}$ is the linear transformation from $\boldsymbol{I}$ to $\boldsymbol{Q}$ for $head_i$, $\boldsymbol{IW}_i^{\boldsymbol{k}}$ is the linear transformation from $\boldsymbol{I}$ to $\boldsymbol{K}$, and $\boldsymbol{IW}_i^{\boldsymbol{V}}$ is the linear transformation from $\boldsymbol{I}$ to $\boldsymbol{V}$. Multi-head attention is calculated by concatenating $head_i$, which is given as follows:

$$MultiHead(\boldsymbol{Q},\ \boldsymbol{K},\ \boldsymbol{V}) = Concat\big(head_1,\ \ldots,\ head_i\big)\boldsymbol{W}^o, \tag{2}$$

where *Concat* refers to tensor concatenation operation, and $\boldsymbol{W}^o$ is a linear transformation matrix. MLP is essentially one fully connected layer involving nonlinear transformations and responsible for adjusting the spatial dimension of feature maps. Meanwhile, normalization can ensure that the network converges faster and is anti-overfitting.

Global and rich contextual information could be captured by Transformer. Placed behind the SPP, Transformer contributes to detect small or occluded objects from complex maritime scenarios.

3.2.3. Network Lightweight Optimization

Computational cost is a strict requirement for drone onboard systems. Based on the premise, how to reduce the algorithm consumption while ensuring its performance becomes a challenge. As one alternative solution, GhostNet is employed in the feature fusion section of the proposed GGT-YOLO.

Let us assume that most feature maps contain redundant information which is similar and ghost-like between one other. The redundant information, called ghost feature maps, guarantees a comprehensive understanding of the input feature map. Using GhostNet, intrinsic and ghost feature maps can be calculated in the following steps. First, m intrinsic feature maps are calculated from input feature maps by convolutions, which is given as follows:

$$\boldsymbol{Y} = \boldsymbol{X} * \boldsymbol{f}, \tag{3}$$

where $\boldsymbol{Y} \in \boldsymbol{R}^{h\prime \times w\prime \times m}$ defines intrinsic feature maps with m channels; $h\prime$ and $w\prime$ are the height and width of $\boldsymbol{Y}$; $\boldsymbol{X} \in \boldsymbol{R}^{h \times w \times c}$ is input feature maps with c channels h and w are the height and width of the input feature map; $\boldsymbol{f} \in \boldsymbol{R}^{c \times k \times k \times m}$ is the convolution filters; and $k \times k$ is the kernel size of $\boldsymbol{f}$. Then, ghost feature maps can be generated by applying a series of cheap linear transformations on each intrinsic feature map in $\boldsymbol{Y}$, as follows:

$$y_{i,j} = \boldsymbol{\Phi}_{i,j}(y_i), \quad \forall\, i = 1, \ldots, m, \quad j = 1, \ldots, S, \tag{4}$$

where y_i is the i-th intrinsic feature map in $\boldsymbol{Y}$, $\boldsymbol{\Phi}_{i,j}$ is the j-th (except the last one) linear transformation, and $y_{i,j}$ is the j-th ghost feature map. S is defined as the number of the generated ghost feature maps. That is to say, each intrinsic feature map y_i can generate one or more ghost feature maps. Finally, both intrinsic and ghost feature maps are combined to form out feature maps. The linear transformations operated on each channel enable a far lesser computational cost to the network than ordinary convolutions. As a result, by using GhostNet the parameters and calculation consumption can be reduced to be about $1/S$ of those of the primary convolution network. S can be considered as the theoretical speed-up ratio of GhostNet.

As shown in Figure 3, two stacked GhostNets and the corresponding shortcuts make up the Ghost bottleneck. One GhostNet acts as an expansion layer to increase the channels of feature maps, while the other one reduces the channels to match the shortcut path. The shortcuts integrate the key information from different layers into the feature maps. Thereby, richer feature information with less computational cost can be obtained by the Ghost bottleneck.

In this work, the Ghost bottleneck is used to replace the CBL in the C3 module, as shown in Figure 3. GhostNet converts intrinsic feature maps to generate ghost feature maps by linear transformations. Compared with the primary network of YOLOv5, floating-point operations and network parameters are greatly reduced. Through comparative

experiments, GhostNet is applied in the last two C3 of our GGT-YOLO algorithm, which is named C3Ghost.

4. Experimental and Discussion

For training and evaluating the proposed GGT-YOLO algorithm, related experiments are performed on a workstation equipped with IntelRCoreTM i7-9800XCPU@3.80GHz ×16, 32 GB RAM and NVIDIA GeForce RTX 2060Ti GPU with 12 GB of memory. Batch size is set as 16, iterations are 300 and the size of the input image is 640 × 640. Other parameters are default. To enhance the diversity of the MariDrone dataset, flip horizontal and mosaic data augmentations are adopted in the phase of training.

4.1. Evaluation Criteria

Detection accuracy and computational cost are the main criteria used to evaluate the performance of the proposed algorithm. In detail, network parameters and floating point operations (FLOPs) are the indicators of computational cost. Precision (P), recall (R) and average precision (AP) are utilized as the indicators of detection accuracy, and the calculations are as follows:

$$\mathrm{P} = \frac{TP}{TP + FP} \tag{5}$$

$$\mathrm{R} = \frac{TP}{TP + FN} \tag{6}$$

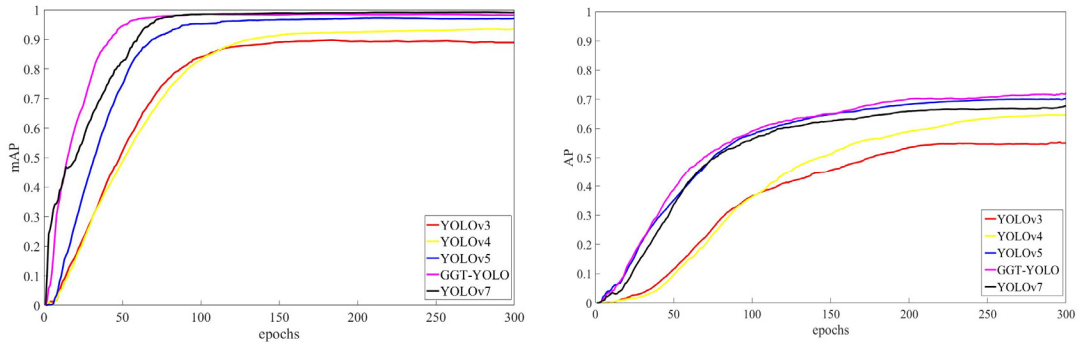
$$\mathrm{AP} = \int_0^1 p(r)dr \tag{7}$$

4.2. Performance Analysis

In this section, the value of the GGT-YOLO algorithm is demonstrated by comparative experiments. The proposed algorithm is compared with YOLOv3 [20], YOLOv4 [23], YOLOv5 [11] and YOLOv7 [28] under same conditions. Not only our MariDrone dataset but also the public dataset RSOD [44] are employed for evaluating the proposed algorithm and other YOLO versions. RSOD is a remote sensing object detection dataset that includes four categories of objects, e.g., aircraft, oil tanks, playground, etc. A total of 976 images and 6950 objects are labeled in the dataset.

Figure 4 shows the APs of these algorithms during training using the RSOD and MariDrone datasets, respectively. It can be seen in Figure 4a that the mean AP (mAP) of GGT-YOLO is 1.0% higher than that of YOLOv5. Although similar to the P, R and mAP of YOLOv7, the parameters and FLOPs of the proposed algorithm are reduced by 83.2% and 85.6%, respectively. In Figure 4b, the APs of all algorithms increase rapidly at the beginning, but the rates gradually slow down when the iteration is about 100. At around 170 epochs, our algorithm GGT-YOLO shows its advantage. During 210 to 250 epochs, GGT-YOLO is almost overlapped with YOLOv5. During the period of 250–300 epochs, all algorithms begin to converge, but GGT-YOLO still maintains high accuracy. As a whole, compared with the YOLO series algorithms, GGT-YOLO has great advantages in convergence speed and accuracy. In addition, it can be noted that the best suitable iteration number is around 300.

To demonstrate the performance of GGT-YOLO and other YOLO series algorithms, P, R, AP, FLOPs and parameters are calculated in Table 1. Compared with YOLOv5, the P of GGT-YOLO is increased by 1.4%, the R is increased by 2.6% and the AP is increased by 1.9%, while its parameters and FLOPs are reduced by 11.6% and 7.3%, respectively. Given these certain advantages, GGT-YOLO is thought more befitting for onboard systems of drones. In addition, the evaluation based on the RSOD dataset is shown in Table 2.



(**a**) (**b**)

Figure 4. The average precisions (APs) of GGT-YOLO and other YOLO versions. (**a**) RSOD dataset; (**b**) MariDrone dataset.

Table 1. Evaluation of GGT-YOLO and other YOLO versions on MariDrone dataset.

Algorithms	P (%)	R (%)	AP (%)	Parameters ($\times 10^6$)	FLOPs ($\times 10^9$)
YOLOv3	64.3	63.6	57.8	61.9	/
YOLOv4	71.0	66.2	65.6	64.1	/
YOLOv5	80.6	69.2	70.2	7.1	16.3
YOLOv7	74.8	70.9	67.0	37.1	104.9
GGT-YOLO	82.0	71.8	72.1	6.2	15.1

Table 2. Evaluation of GGT-YOLO and other YOLO versions on RSOD dataset.

Algorithms	P (%)		R (%)		mAP (%)
	Aircraft	Oiltank	Aircraft	Oiltank	
YOLOv3	91.9	86.1	90.7	85.5	88.3
YOLOv4	96.4	92.6	94.3	86.2	93.9
YOLOv5	98.7	95.3	95.9	93.7	96.5
YOLOv7	99.6	96.8	97.5	98.1	98.7
GGT-YOLO	98	96.2	96.7	97.4	97.5

4.3. Comparative Analysis

During maritime cruising executed by drones, ships in the remote and moving view present scale variations and frequent occlusions. Aside for computational cost, detection accuracy is also required by the onboard vision detection algorithm. As described in Section 3, Transformer is introduced to enhance the feature extraction of YOLOv5, while GhostNet is introduced to reduce the computational cost. How to fuse the two models with the primary network is analyzed in this section.

The proposed GGT-YOLO and other derivatives are defined in Table 3. Bn represents the *n*-th C3 model behind SPP, where GhostNet or Transformer is introduced. GGT-YOLO is defined by one Transformer being used to replace the first C3 model and two GhostNets being used to replace the fourth and fifth C3 models in the YOLOv5 framework. T-YOLO is defined by one Transformer being used to replace the first C3 model in the framework of YOLOv5. G-YOLO is defined by one GhostNet being used to replace the fifth C3 model in the framework of YOLOv5. GT-YOLO is defined by one Transformer being used to replace the first C3 model and one GhostNet being used to replace the fifth C3 model in the framework of YOLOv5. Other derivatives, e.g., TT-YOLO, GG-YOLO and GGGT-YOLO, are also defined in the same way. By comparative experiments between these derivatives, the optimal solution for drone-based maritime object detection can be obtained.

Table 3. Definitions of various modification versions based on Transformer and C3Ghost.

Model	B1	B2	B3	B4	B5
G-YOLO					GhostNet
GG-YOLO				GhostNet	GhostNet
T-YOLO	Transformer				
TT-YOLO	Transformer	Transformer			
GT-YOLO	Transformer				GhostNet
GGT-YOLO	**Transformer**			**GhostNet**	**GhostNet**
GGGT-YOLO	Transformer		GhostNet	GhostNet	GhostNet

The APs of these fresh networks designed by tentative combination are calculated in Figure 5. It can be seen that even though all the networks have converged, GGT-YOLO proposed by our work has a faster rise speed in the beginning stage and keeps a higher score in the final stage. In addition, the corresponding evaluation metrics are listed in Table 4, and our GGT-YOLO is highlighted in bold.

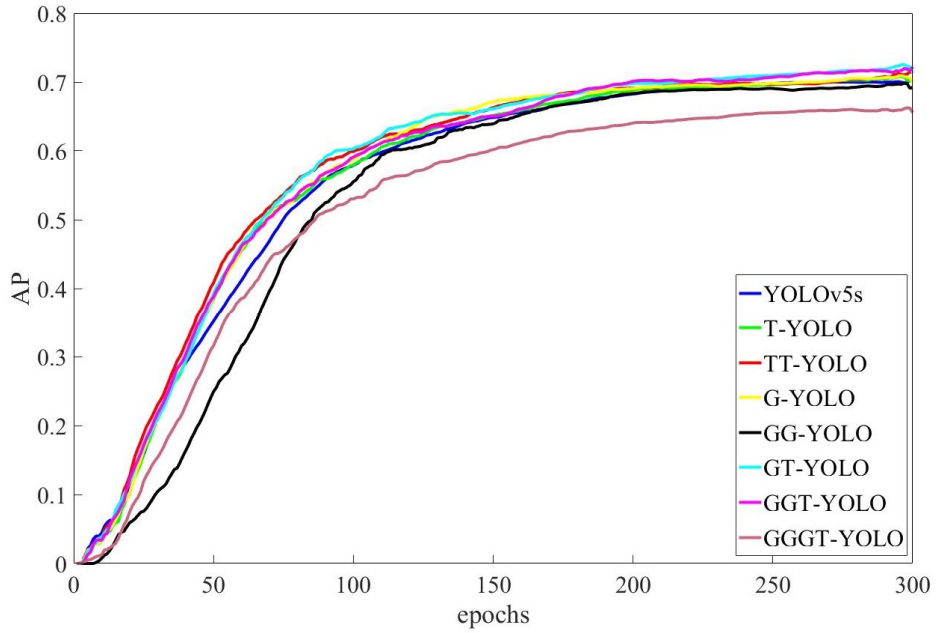


Figure 5. The APs of GGT-YOLO and other modification versions.

Table 4. Performance of GGT-YOLO and other algorithms on MariDrone dataset.

Model	P (%)	R (%)	AP (%)	Parameters ($\times 10^6$)	FLOPs (10^9)
YOLOv5	80.6	69.2	70.2	7.05	16.3
T-YOLO	81.3	71.6	71.8	7.05	16.1
TT-YOLO	84.4	67.5	71.9	7.06	15.9
G-YOLO	83.3	70.9	70.4	6.40	15.8
GG-YOLO	77.4	68.5	69.6	6.23	15.3
GT-YOLO	82.0	69.6	72.4	6.40	15.6
GGT-YOLO	**82.0**	**71.8**	**72.1**	**6.23**	**15.1**
GGGT-YOLO	82.0	64.7	66.7	6.19	14.5

Owing to one C3Ghost applied in the neck section, the parameters and FLOPs of G-YOLO are reduced by 0.6×10^6 and 0.5×10^9, respectively. Meanwhile, the AP remains at the same level as YOLOv5. To further investigate whether C3Ghost has an effect on reducing computational cost, GG-YOLO (that applies two C3Ghost models) is studied.

As shown in Table 4, even though the computational cost is less than for G-YOLO, the AP of GG-YOLO starts to decrease. It shows that GhostNet in C3Ghost would affect the detection accuracy when reducing computation complexity. To guarantee a reliable detection accuracy, another T-YOLO introduces one Transformer in the backbone section of YOLOv5. The results in the Table 3 show that the P, R and AP of T-YOLO are improved by 0.7%, 2.4% and 0.6%, respectively. Unfortunately, when two Transformer models are introduced into the network, the AP is improved by only 0.1%, but there is a decrease of 1.7% in R. It shows that Transformer could improve the average detection precision, but the recall would not.

In order to better balance computational cost and detection accuracy, a novel GGT-YOLO algorithm is found to be the optimal solution according to comparisons of the evaluation metrics. One Transformer and two C3Ghost models are introduced in the GGT-YOLO. For proof, another two networks, GT-YOLO and GGGT-YOLO, are also designed (in Table 3). GT-YOLO replaces one C3 with one C3Ghost in the neck and introduces one Transformer in the backbone. Even though the detection accuracy of GGT-YOLO is the same as that of GT-YOLO, the parameters and FLOPs are fewer. This means that GGT-YOLO has a lower computational cost. Furthermore, GGGT-YOLO applies more C3Ghost models and is compared with GGT-YOLO; the detection accuracy degenerates rapidly, though a lesser computational complexity is available. As showed in Figure 5, GGGT-YOLO does not seem to perform well in the convergence stage.

4.4. Results and Discussion

Thousands of images were recorded when drones were implementing the mission of maritime cruise. Various situations are involved in the dataset, e.g., single object, multi-object, sunny, cloudy, etc. Different sizes and orientation of ships are also presented with labels in these images. Through training, GGT-YOLO is tested and evaluated by using the testing set and validation set. Part of the results are shown in Figure 6. It can be seen that all ships, including small or occluded ships, are detected from large-scale crowded backgrounds.

By the exploratory experiments above, an optimal algorithm GGT-YOLO is proposed for drone-based vision to detect ships from maritime scenarios. Considering the limited computational ability of the onboard system, GhostNet is introduced to reduce the proposed algorithm's computational cost. Instead of general convolution calculation, linear transformations are employed to generate feature maps in GhostNet, and fewer FLOPs are required. It is beneficial for the proposed algorithm to be deployed on airborne systems. However, as more GhostNet models are introduced, the detection accuracy involving P, R and AP begins to decrease. The reason is that linear transformations of GhostNet can not fully approximate the convolution operation. On the other hand, Transformer is proved to have the ability to enhance the detection accuracy of the algorithm. The multi-head attention is able to calculate the contexts of pixels in different positions from multi- subspaces, which is beneficial for GGT-YOLO in extracting significant features from large-scale scenarios.

In conclusion, lesser computational cost as well as adequate detection accuracy has been achieved by our GGT-YOLO. The corresponding P, R and AP are 82%, 71.8% and 72.1%, respectively. In addition, the parameters and FLOPs are 6,234,710 and 15×10^9. Through the comparative experiments, it can be noted that proper introduction of Transformer and GhostNet is beneficial to improve the performance of the detection algorithm. The proposed GGT-YOLO is available for detecting maritime objects by drones.

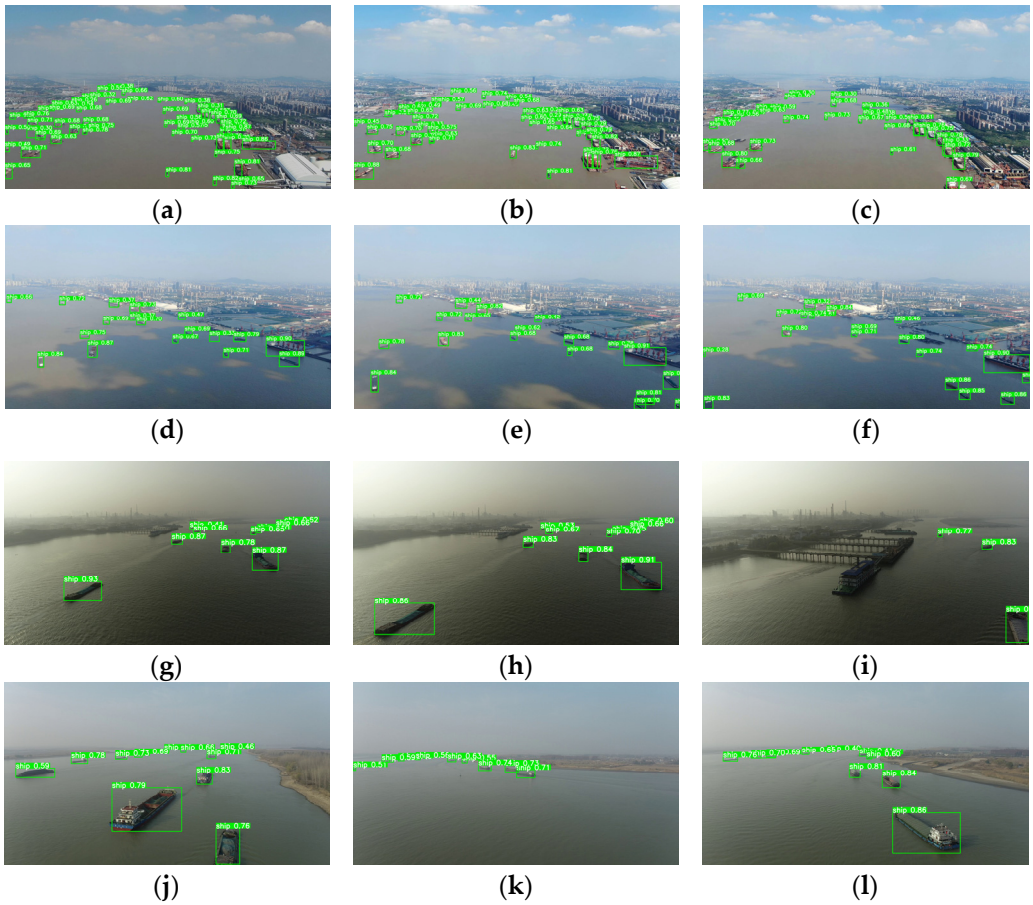


Figure 6. Quantitative detection results. (**a**–**c**) scenario 1: sunny, dense; (**d**–**f**) scenario 2: sunny, dense; (**g**–**i**) scenario 3: cloudy, occluded; (**j**–**l**) scenario 4: sunny, occluded.

5. Conclusions

Both detection accuracy and computational consumption require consideration simultaneously when drones are being employed to detect small or occluded objects from large-scale scenarios. In this work, we proposed a novel drone-based maritime object detection algorithm, in which the feature extraction is enhanced while the computation of the feature fusion is optimized. A specialized dataset is introduced, and numerous comparative experiments have been conducted to illustrate the proposed algorithm. The results show that the P, R and AP are improved by 1.4%, 2.6% and 1.9%, respectively, compared with the primary YOLOv5. Furthermore, the parameters and floating-point operations are reduced by 11.6% and 7.3%, respectively. It can be proved that the algorithm provides a single optimal solution for drone-based object detection in maritime and other remote sensing fields. In the next work, the lightweight of the feature fusion will be studied.

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

Funding: This research was funded by National Natural Science Foundation of China, grant number 52001235; China Postdoctoral Science Foundation, grant number 2020M682504; Shandong Provincial Natural Science Foundation, grant number ZR2020KE029; Scientific Research Project of Hubei Education Department, grant number D20201501); Youth Science Foundation of WIT, grant number 000017/19QD07.

Data Availability Statement: Not applicable.

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

References

1. Drone Industry Insights. Global Drone Market Report 2022–2030. Available online: https://droneii.com/ (accessed on 23 October 2022).
2. Liu, Y.; Liu, H.; Tian, Y.; Sun, C. Reinforcement Learning Based Two-level Control Framework of UAV Swarm for Cooperative Persistent Surveillance in an Unknown Urban Area. *Aerosp. Sci. Technol.* **2020**, *98*, 105671. [CrossRef]
3. Yuan, H.; Xiao, C.; Wang, Y.; Peng, X.; Wen, Y.; Li, Q. Maritime Vessel Emission Monitoring by An UAV Gas Sensor System. *Ocean Eng.* **2020**, *218*, 105206. [CrossRef]
4. Jeong, G.Y.; Nguyen, T.N.; Tran, D.K.; Hoang, T.B.H. Applying Unmanned Aerial Vehicle Photogrammetry for Measuring Dimension of Structural Elements in Traditional Timber Building. *Measurement* **2020**, *153*, 107386. [CrossRef]
5. Zhang, X.; Zhao, P.; Hu, Q.; Ai, M.; Hu, D.; Li, J. A UAV-based Panoramic Oblique Photogrammetry (POP) Approach Using Spherical Projection. ISPRS J. Photogramm. *Remote Sens.* **2020**, *159*, 198–219.
6. Yuan, H.; Xiao, C.; Zhan, W.; Wang, Y.; Shi, C.; Ye, H.; Jiang, K.; Ye, Z.; Zhou, C.; Wen, Y.; et al. Target Detection, Positioning and Tracking Using New UAV Gas Sensor Systems: Simulation and Analysis. *J. Intell. Robot. Syst.* **2019**, *94*, 871–882. [CrossRef]
7. Zou, Z.; Shi, Z.; Guo, Y.; Ye, J. Object Detection in 20 Years: A Survey. *arXiv* **2019**, arXiv:1905.05055. [CrossRef]
8. Li, W.; Li, F.; Luo, Y.; Wang, P. A Survey of Deep Learning-Based Object Detection. *IEEE Access* **2019**, *7*, 128837–128868.
9. VisDrone Dataset. Available online: http://aiskyeye.com/download/ (accessed on 23 October 2022).
10. Okutama-Action Dataset. Available online: https://github.com/miquelmarti/Okutama-Action (accessed on 23 October 2022).
11. Ultralytics. YOLOv5. Available online: https://github.com/ultralytics/yolov5 (accessed on 1 November 2020).
12. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Houlsby, N. An Image is Worth 16 × 16 words: Transformers for Image Recognition at Scale. *arXiv* **2020**, arXiv:2010.11929. [CrossRef]
13. Han, K.; Wang, Y.; Tian, Q.; Guo, J.; Xu, C. GhostNet: More Features from Cheap Operations. *arXiv* **2020**, arXiv:1911.11907. [CrossRef]
14. Zhang, Z.; Liu, Y.; Liu, T.; Lin, Z.; Wang, S. DAGN: A Real-Time UAV Remote Sensing Image Vehicle Detection Framework. *IEEE Geosci. Remote Sens. Lett.* **2020**, *17*, 1884–1888. [CrossRef]
15. Senthilnath, J.; Varia, N.; Dokania, A.; Anand, G.; Benediktsson, J.A. Deep TEC: Deep Transfer Learning with Ensemble Classifier for Road Extraction from UAV Imagery. *Remote Sens.* **2020**, *12*, 245. [CrossRef]
16. Zhu, J.; Zhong, J.; Ma, T.; Huang, X.; Zhang, W.; Zhou, Y. Pavement Distress Detection Using Convolutional Neural Networks with Images Captured via UAV. *Autom. Constr.* **2022**, *133*, 103991. [CrossRef]
17. Liu, Y.; Yang, F.; Hu, P. Small-Object Detection in UAV-Captured Images via Multi-Branch Parallel Feature Pyramid Networks. *IEEE Access* **2020**, *8*, 145740–145750. [CrossRef]
18. Liu, J.; Wang, Z.; Wu, Y.; Qin, Y.; Cao, X.; Huang, Y. An Improved Faster R-CNN for UAV-Based Catenary Support Device Inspection. *Int. J. Softw. Eng. Knowl. Eng.* **2020**, *30*, 941–959. [CrossRef]
19. Sun, W.; Dai, L.; Zhang, X.; Chang, P.; He, X. RSOD: Real-time small object detection algorithm in UAV-based traffic monitoring. *Appl. Intell.* **2022**, *52*, 8448–8463. [CrossRef]
20. Redmon, J.; Farhadi, A. YOLOv3: An Incremental Improvement. *arXiv* **2018**, arXiv:1804.02767.
21. Liu, M.; Wang, X.; Zhou, A.; Fu, X.; Ma, Y.; Piao, C. UAV-YOLO: Small Object Detection on Unmanned Aerial Vehicle Perspective. *Sensors* **2020**, *20*, 2238. [CrossRef]
22. Tan, L.; Lv, X.; Lian, X.; Wang, G. YOLOv4_Drone: UAV Image Target Detection Based on An Improved YOLOv4 Algorithm. *Comput. Electr. Eng.* **2021**, *93*, 107261. [CrossRef]
23. Bochkovskiy, A.; Wang, C.Y.; Liao, H.Y.M. YOLOv4: Optimal Speed and Accuracy of Object Detection. *arXiv* **2020**, arXiv:2004.10934.
24. Zhan, W.; Sun, C.; Wang, M.; She, J.; Zhang, Y.; Zhang, Z.; Sun, Y. An Improved Yolov5 Real-time Detection Method for Small Objects Captured by UAV. *Soft. Comput.* **2022**, *26*, 361–373. [CrossRef]
25. Zhao, J.; Zhang, X.; Yan, J.; Qiu, X.; Yao, X.; Tian, Y.; Zhu, Y.; Cao, W. A Wheat Spike Detection Method in UAV Images Based on Improved YOLOv5. *Remote Sens.* **2021**, *13*, 3095. [CrossRef]
26. Chen, G.; Wang, H.; Chen, K.; Li, Z.; Song, Z.; Liu, Y.; Chen, W.; Knoll, A. A Survey of the Four Pillars for Small Object Detection: Multiscale Representation, Contextual Information, Super-Resolution, and Region Proposal. *IEEE Trans. Syst. Man. Cybern. Syst.* **2022**, *52*, 936–953. [CrossRef]
27. Li, C.; Li, L.; Jiang, H.; Weng, K.; Geng, Y.; Li, L.; Ke, Z.; Li, Q.; Cheng, M.; Nie, W.; et al. YOLOv6: A Single-Stage Object Detection Framework for Industrial Applications. *arXiv* **2022**, arXiv:2209.02976. [CrossRef]
28. Wang, C.; Bochkovskiy, A.; Liao, H.M. YOLOv7: Trainable Bag-of-freebies Sets New State-of-the-art for Real-time Object Detectors. *arXiv* **2022**, arXiv:2207.02696. [CrossRef]
29. Liu, Y.; Cao, S.; Lasang, P.; Shen, S. Modular Lightweight Network for Road Object Detection Using a Feature Fusion Approach. *IEEE Trans. Syst. Man. Cybern. Syst.* **2021**, *51*, 4716–4728. [CrossRef]
30. Lv, Y.; Liu, J.; Chi, W.; Chen, G.; Sun, L. An Inverted Residual Based Lightweight Network for Object Detection in Sweeping Robots. *Appl. Intell.* **2022**, *52*, 12206–12221. [CrossRef]

31. Javadi, S.; Dahl, M.; Pettersson, M.I. Vehicle Detection in Aerial Images Based on 3D Depth Maps and Deep Neural Networks. *IEEE Access* **2021**, *9*, 8381–8391. [CrossRef]
32. Li, D.; Sun, X.; Elkhouchlaa, H.; Jia, Y.; Yao, Z.; Lin, P.; Li, J.; Lu, H. Fast Detection and Location of Longan Fruits Using UAV Images. *Comput. Electron. Agric.* **2021**, *190*, 106465. [CrossRef]
33. Kou, M.; Zhou, L.; Zhang, J.; Zhang, H. Research Advances on Object Detection in Unmanned Aerial Vehicle Imagery. *Meas. Control Technol.* **2020**, *39*, 47–61.
34. Prasad, D.K.; Prasath, C.K.; Rajan, D.; Rachmawati, L.; Rajabally, E.; Quek, C. Object Detection in A Maritime Environment: Performance Evaluation of Background Subtraction Methods. *IEEE Trans. Intell. Transp. Syst.* **2019**, *20*, 1787–1802. [CrossRef]
35. Prasad, D.K.; Dong, H.; Rajan, D.; Chai, Q. Are Object Detection Assessment Criteria Ready for Maritime Computer Vision? *IEEE Trans. Intell. Transp. Syst.* **2020**, *21*, 5295–5304. [CrossRef]
36. Shao, Z.; Wu, W.; Wang, Z.; Wan, D.; Li, C. SeaShips: A Large-Scale Precisely Annotated Dataset for Ship Detection. *IEEE Trans. Multimed.* **2018**, *20*, 2593–2604. [CrossRef]
37. Iancu, B.; Soloviev, V.; Zelioli, L.; Lilius, J. ABOships-An Inshore and Offshore Maritime Vessel Detection Dataset with Precise Annotations. *Remote Sens.* **2021**, *13*, 988. [CrossRef]
38. Gallegos, A.; Pertusa, A.; Gil, P.; Fisher, R.B. Detection of Bodies in Maritime Rescue Operations using Unmanned Aerial Vehicles with Multispectral Cameras. *J. Field Robot.* **2019**, *36*, 782–796. [CrossRef]
39. Liu, T.; Pang, B.; Zhang, L. Sea Surface Object Detection Algorithm Based on YOLOv4 Fused with Reverse Depth wise Separable Convolution (RDSC) for USV. *J. Mar. Sci. Eng.* **2021**, *9*, 753. [CrossRef]
40. Ghahremani, A.; Alkanat, T.; Bondarev, E.; de With, P.H.N. Maritime vessel Re-identification: Novel VR-VCA dataset and a Multi-branch Architecture MVR-net. *Mach. Vis. Appl.* **2021**, *32*, 71. [CrossRef]
41. Li, Y.; Zhang, S.; Wang, W. A Lightweight Faster R-CNN for Ship Detection in SAR Images. *IEEE Geosci. Remote Sens. Lett.* **2022**, *19*, 4006105. [CrossRef]
42. Nie, T.; Han, X.; He, B.; Li, X.; Liu, H.; Bi, G. Ship Detection in Panchromatic Optical Remote Sensing Images Based on Visual Saliency and Multi-Dimensional Feature Description. *Remote Sens.* **2020**, *12*, 152. [CrossRef]
43. Guo, H.; Yang, X.; Wang, N.; Gao, X. A CenterNet Plus Plus model for Ship Detection in SAR Images. *Pattern Recognit.* **2021**, *112*, 107787. [CrossRef]
44. Long, Y.; Gong, Y.; Xiao, Z.; Liu, Q. Accurate Object Localization in Remote Sensing Images Based on Convolutional Neural Networks. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 2486–2498. [CrossRef]