electronics

MDPI

Article

MegaDetectNet: A Fast Object Detection Framework for Ultra-High-Resolution Images

Jian Wang [1,2,*], Yuesong Zhang [1], Fei Zhang [1], Yazhou Li [1], Lingcong Nie [1] and Jiale Zhao [1]

[1] Faculty of Information Engineering and Automation, Kunming University of Science and Technology, Kunming 650504, China; 20212204295@stu.kust.edu.cn (Y.Z.); 20222104055@stu.kust.edu.cn (F.Z.); 20212204293@stu.kust.edu.cn (Y.L.); 20212204158@stu.kust.edu.cn (L.N.); jiale_work@163.com (J.Z.)

[2] Yunnan Key Lab of Artificial Intelligence, Kunming University of Science and Technology, Kunming 650504, China

* Correspondence: jianwang@kust.edu.cn; Tel.: +86-130-9942-4960

Abstract: Addressing the challenge of efficiently detecting objects in ultra-high-resolution images during object detection tasks, this paper proposes a novel method called MegaDetectNet, which leverages foreground image for large-scale resolution image object detection. MegaDetectNet utilizes a foreground extraction network to generate a foreground image that highlights target regions, thus avoiding the computationally intensive process of dividing the image into multiple sub-images for detection, and significantly improving the efficiency of object detection. The foreground extraction network in MegaDetectNet is built upon the YOLOv5 model with modifications: the large object detection head and classifier are removed, and the PConv convolution is introduced to reconstruct the C3 module, thereby accelerating the convolution process and enhancing foreground extraction efficiency. Furthermore, a Res2Rep convolutional structure is developed to enlarge the receptive field and improve the accuracy of foreground extraction. Finally, a foreground image construction method is proposed, fusing and stitching foreground target regions into a unified foreground image. This approach replaces multiple divided sub-images with a single foreground image for detection, reducing overhead time. The proposed MegaDetectNet method's effectiveness for detecting objects in ultra-high-resolution images is validated using the publicly available DOTA dataset. Experimental results demonstrate that MegaDetectNet achieves an average time reduction of 83.8% compared to the sub-image division method among various commonly used object detectors, with only a marginal 8.7% decrease in mAP (mean Average Precision). This validates the practicality and efficacy of the MegaDetectNet method for object detection in ultra-high-resolution images.

Keywords: ultra-high-resolution images; object detection; foreground extraction network; foreground image

Citation: Wang, J.; Zhang, Y.; Zhang, F.; Li, Y.; Nie, L.; Zhao, J. MegaDetectNet: A Fast Object Detection Framework for Ultra-High-Resolution Images. *Electronics* **2023**, *12*, 3737. https://doi.org/10.3390/electronics12183737

Academic Editor: Seong G. Kong

Received: 8 August 2023
Revised: 28 August 2023
Accepted: 31 August 2023
Published: 5 September 2023

1. Introduction

With the continuous advancement of photography equipment and image acquisition technology, obtaining ultra-high-resolution images has become more accessible. These high-resolution images find extensive applications in various domains, such as remote sensing, medical imaging, and surveillance. However, due to their complexities and computational demands, fast object detection in ultra-high-resolution images remains challenging. Directly inputting such large images into detection models may lead to losing crucial details after multiple downsampling steps, making it challenging to detect numerous small objects. Conversely, it would require substantial memory resources during forward propagation to reduce the downsampling frequency, potentially exhausting GPU resources and hindering proper training and inference. In light of this, an effective solution to address the object detection problem in ultra-high-resolution images involves reducing the image resolution before detection.

Several methods for object detection in large-resolution images have been proposed. Some approaches involve dividing the image into multiple sub-images for detection [1–5], while others employ multiscale fusion strategies to detect features at different resolutions [6–8]. Additionally, attention mechanisms have been incorporated to enhance detection performance by focusing on regions of interest [9–11]. However, both multiscale strategies and attention mechanisms encounter challenges when applied to ultra-high-resolution images, as striking a balance between detection accuracy and GPU memory resources becomes intricate. While these strategies are effective with aerial remote sensing datasets of resolutions below 2000 × 2000, their efficacy diminishes when dealing with larger image resolutions, as exemplified by the DOTA dataset [12] (with a maximum image resolution of 13,383 × 4287). Additionally, subdividing images into sub-images, though advantageous for enhancing detection precision, brings about substantial computational overhead in the case of object detection in ultra-high-resolution images. The generation and retention of numerous sub-images incur storage and time-related costs.

Ultra-high-resolution images typically exhibit low foreground-to-background ratios, as evident in the DOTA dataset, with an average foreground ratio of only 0.042 [12]. For images with resolutions of 10,000 × 10,000, the optimal size of the foreground image is only 420 × 420, which can be directly input into the object detection model. Based on this characteristic, this paper proposes a novel method for object detection in ultra-high-resolution images. The approach involves utilizing a foreground extraction network to obtain approximate target regions of the image, which are then appropriately scaled and stitched to form a single foreground image for detection. The proposed MegaDetectNet converts the challenging task of detecting ultra-high-resolution images into foreground images by eliminating redundant background information. On the DOTA dataset, which contains diverse high-resolution aerial remote sensing data, MegaDetectNet achieves an average time reduction of 83.8% compared to mainstream sliding window cropping methods used in the field of super-resolution object detection, with only a mere 8.7% decrease in mAP.

The primary contributions of this paper are as follows:

1. A foreground extraction network is designed based on a YOLOv5 [13] model of size S in version 6.0. This network eliminates the large object detection head and target classification structure. It reconstructs the C3 feature extraction module using the partial convolution (PConv) structure, thus reducing the foreground extraction network's model complexity and computational load while effectively enhancing its efficiency.
2. A Res2Rep convolutional structure for multiscale feature extraction is constructed using a cascading residual network and reparameterization method, effectively enlarging the foreground extraction network's receptive field and enhancing the precision of foreground extraction.
3. A novel greedy strategy for generating foreground images is introduced. This strategy involves aggregating and extending the results of foreground extraction to delineate foreground regions, which are then scaled and concatenated to construct the foreground images. The approach circumvents the resource overhead of recurrently detecting numerous sub-images by shifting from sub-image detection in ultra-high-resolution images to foreground image-based detection. Simultaneously, this strategy ensures the precision of object detection in ultra-high-resolution images by accommodating various foreground area sizes.

2. Related Work

2.1. Object Detection

Object detection involves locating objects of interest in images, determining their categories, and providing bounding box position information. Deep learning-based object detection algorithms can be categorized into two-stage and one-stage algorithms. Two-stage algorithms generate and refine candidate regions to obtain detection results, achieving

higher accuracy but slower detection speeds (e.g., Faster RCNN [14] and Mask RCNN [15]). One-stage algorithms directly provide detection results without candidate generation, offering faster detection but potentially lower accuracy (e.g., YOLO [16,17], RetinaNet [18], SSD [19], and FCOS [20]).

The foreground region used in this paper resembles the candidate regions in two-stage algorithms. Prior works [21–25] on object detection have utilized foreground regions or region-of-interest approaches, designing efficient and accurate foreground detectors. However, these methods solely perform pre-localization for each object position to ensure more precise detection results without applying foreground regions to address challenges in ultra-high-resolution image object detection and rapid foreground region extraction tasks.

2.2. Object Detection in Ultra-High-Resolution Images

Object detection in satellite remote sensing and aerial imagery is a significant application area of current object detection technologies. However, these images typically possess enormous resolutions (e.g., $10,000 \times 10,000$), making direct input into detection models challenging. Moreover, most objects in these images are tiny (e.g., 10×10) and often clustered together, making recognition difficult.

Currently, three main approaches address object detection in large-resolution images include: sub-image division, multiscale strategies, and attention-based methods.

The sub-image division approach, initially proposed by Van et al. [1], involves dividing the ultra-high-resolution image into multiple sub-images and merging the detection results of each sub-image to obtain the final detection results for the entire image. This approach includes sliding window cropping and random center point cropping. Random center point cropping segments sub-images based on the center points of object bounding boxes, resulting in many duplicated sub-images when objects are densely distributed, making it less suitable for detecting objects in ultra-high-resolution images. Sliding window cropping has been widely used in object detection tasks for ultra-high-resolution images, typically employing a crop size of 1024 [2–4]. However, the sub-image division approach may encounter issues such as missed or duplicated object detections when objects span across sub-image boundaries, and high computational and memory costs, making it unsuitable for fast object detection scenarios.

Multiscale strategies involve scaling the image to different resolutions to create an image pyramid, extracting features from each scale, and performing object detection using features from each scale. SNIP [26] proposes detecting small objects with enlarged feature maps and large objects with downscaled feature maps to reduce resource consumption in feature detection. PANet [27] enhances the feature pyramid by incorporating bidirectional connections between scales. RDN [7] and ThunderNet [28] merge features from different scales to reduce computation costs. GiraffeDet [29] introduces cross-scale connections to fuse features from the previous and current layers to cope with extreme scale variations. QueryDet [30] predicts the rough positions of small objects on low-resolution features and computes accurate positions on high-resolution features.

Attention-based methods leverage spatial or channel-wise attention mechanisms to weigh the input features and enhance their perception capabilities in spatial and channel dimensions to recognize small objects in large-resolution images. To detect subtle small objects in large-resolution images, MViTv2 [31] proposes a pooling attention mechanism, DESTR [31] introduces cross-attention for both classification and regression branches, and Zhu et al. [32] propose a bidirectional cross-attention mechanism.

However, both multiscale strategies and attention-based methods have limited capabilities in handling ultra-high-resolution images, demonstrating excellent performance at relatively high resolutions (e.g., 2000×2000) but still needing help to handle even larger resolution images (e.g., $10,000 \times 10,000$).

3. Methodology

This framework aims to enhance the detection performance of object detection models on ultra-high-resolution images. As described in Figure 1, the detection process involves using a foreground extraction network to generate a foreground image and subsequently performing object detection on this map using a general detector. The detection results are returned to the original ultra-high-resolution image to obtain the final detections.

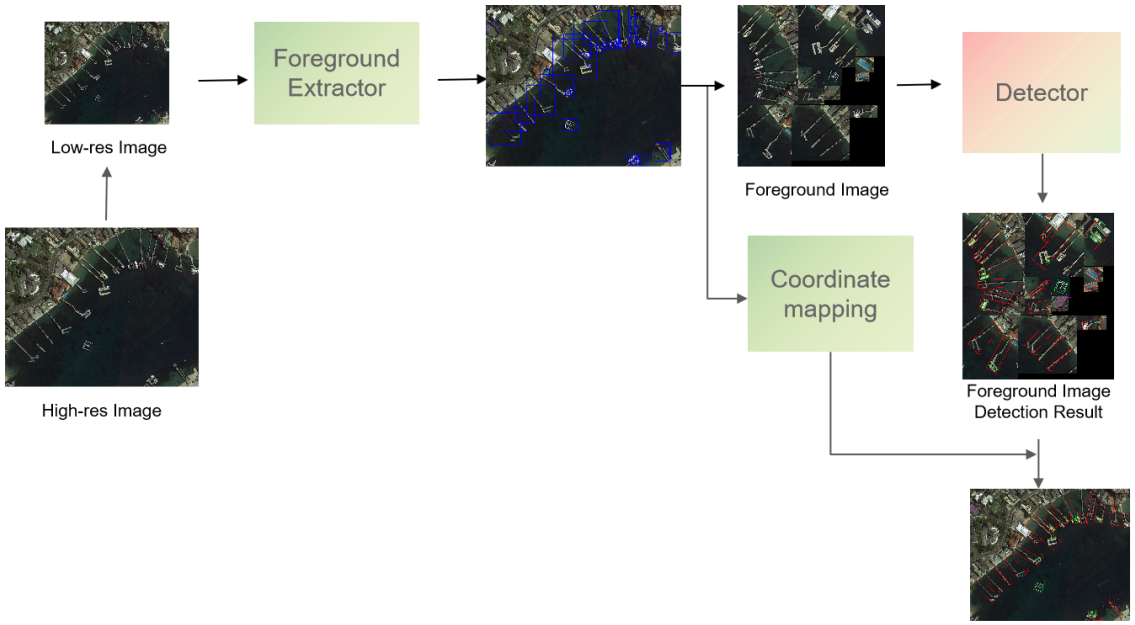


Figure 1. Pipeline of MegaDetectNet.

3.1. Foreground Extraction Network

We designed a network based on YOLOv5 version 6.0 [13] to ensure efficient foreground extraction, as illustrated in Figure 2. The foreground extraction network consists of three main components: Backbone, Neck, and Head. The Backbone component is responsible for extracting image features, the Neck component fuses shallow and deep features to obtain comprehensive feature representations, and the Head component focuses on detecting target positions. Both the CBS and SimSPPF modules are integral constituents of the YOLO series algorithms. The CBS module embodies a fundamental convolutional structure encompassing convolutional layers, batch normalization layers, and activation functions. Meanwhile, the SimSPPF module is a spatial pyramid module designed to facilitate multi-scale feature fusion.

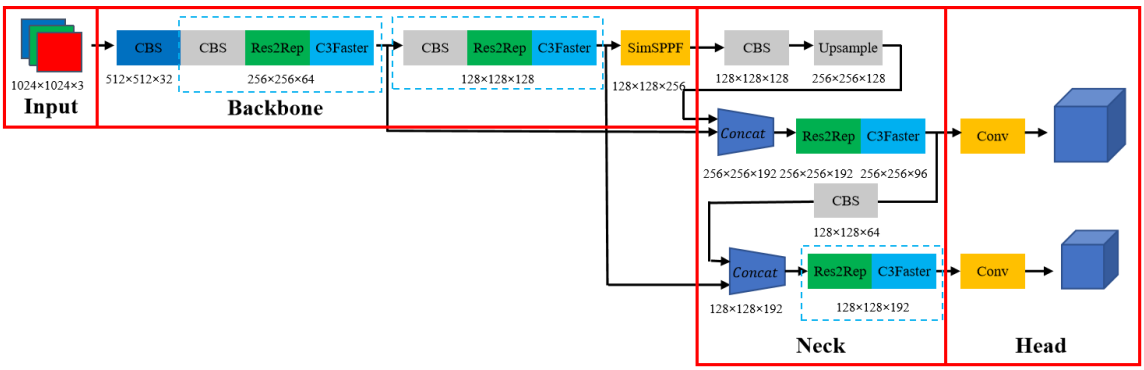


Figure 2. Architecture of the foreground extraction network. The blue square represents the detection head of the foreground extraction network. Below each module, the shapes of the respective output feature maps are indicated.

To address the challenges of foreground region extraction in ultra-high-resolution images, we made the following design choices:

1. We removed the detection head and classifier for large objects to reduce computational complexity and improve foreground region extraction efficiency.
2. To further enhance the extraction efficiency, we introduced the C3Faster module, which uses the PConv convolution structure to achieve rapid feature map dimension reduction.
3. To effectively increase the receptive field and enhance foreground region extraction accuracy, we designed the Res2Rep convolution structure based on the Res2Net network.

3.1.1. Removal of Large Object Detection Head and Classifier

Ultra-high-resolution images have enormous object sizes, and objects are often clustered together, as shown in Figure 3. After multiple downsampling operations, large objects become medium-sized, and small objects may become difficult to detect or even disappear. The detection head for large objects struggles to detect targets. When the foreground extraction network detects medium-sized and small objects, the merged sub-regions already encompass most small objects. Expanding the object regions outward from their center points ensures that the foreground image includes more small objects. Since small objects are usually clustered with large or medium-sized objects in ultra-high-resolution images, appropriately enlarging the foreground regions ensures the coverage of most objects. Therefore, we removed the large object detection head from YOLOv5 and retained the detection heads for medium-sized and small objects, as shown in Figure 2.

Figure 3. Some ultra-high-resolution images from the DOTA dataset.

After removing the large object detection head, we significantly reduced the model size of the foreground extraction network, saving computational resources while covering most small objects in the foreground regions. Additionally, since foreground extraction only requires object localization without object classification, we removed the classifier to improve foreground extraction efficiency.

3.1.2. C3Faster Structure

In YOLOv5, the C3 module is the primary module for learning residual features. It consists of two branches: one with multiple stacked bottlenecks and three basic convolution modules, and the other with a single basic convolution module. The outputs from both branches are concatenated. The bottleneck module, composed of two convolutional layers, one 1×1 convolutional layer and one 3×3 convolutional layer, reduces parameter count while deepening the network, resulting in faster computational speeds.

Research [33–35] indicates that features from different channels exhibit high similarity. A new convolution structure called PConv (Partial Convolution) was introduced [33], which

addresses the challenge of high computational complexity in conventional convolution (Conv) by leveraging the redundant characteristics of feature maps. It applies conventional convolution only to some input channels, effectively representing the entire feature map with reduced FLOPs. Typically, PConv achieves similar effectiveness as conventional convolution but with only 1/4 of the FLOPs. Experimental results [33] demonstrate the effectiveness of PConv in extracting spatial features.

Considering the imperative for a faster feature extraction process and lighter model weights within the foreground extraction network, we have designed the FasterBlock module based on the PConv convolutional structure. This module aims to replace the bottleneck section within the C3 module, giving rise to the C3Faster architecture, as depicted in Figure 4.

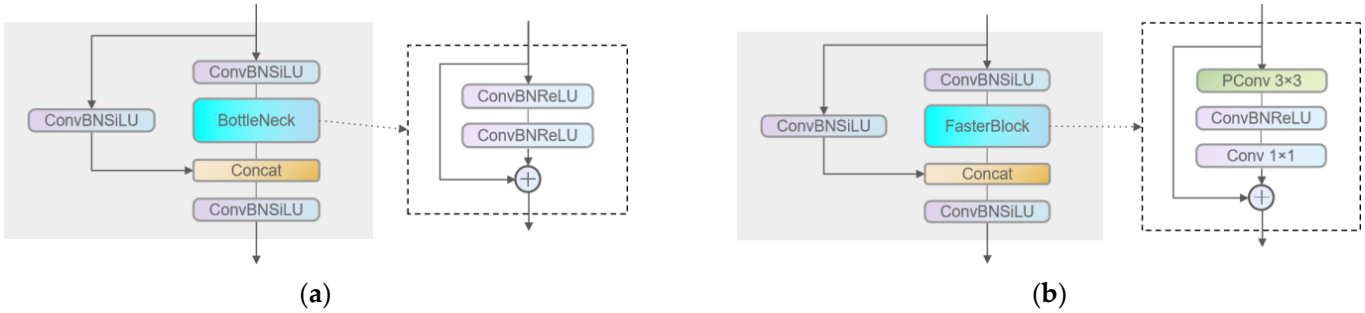


Figure 4. Comparison of C3 and C3Faster architectures. (**a**) C3 structure; (**b**) C3Faster structure.

Specifically, the PConv convolution performs a 3×3 convolution on input feature maps with a channel ratio 1/4. The convolution result is then concatenated with the remaining 3/4 feature channels, yielding the convolutional output of PConv. We have omitted the second non-linear transformation within the FasterBlock module to alleviate the computational burden, retaining only the convolution operation. By enhancing the feature extraction capability of the FasterBlock module without compromising detection accuracy, we have successfully accelerated detection speed while diminishing computational overhead.

3.1.3. Res2Rep Structure

The RepVGG [36] structure is a reparameterized architecture with multiple branches during training, which can be merged into a single 3×3 convolution during deployment, as shown in Figure 5. RepVGG effectively utilizes the computational power of GPUs through 3×3 convolutional structures. Experiments show that YOLOv6 [37] reduces hardware delays and significantly improves the algorithm's precision by adopting RepVGG.

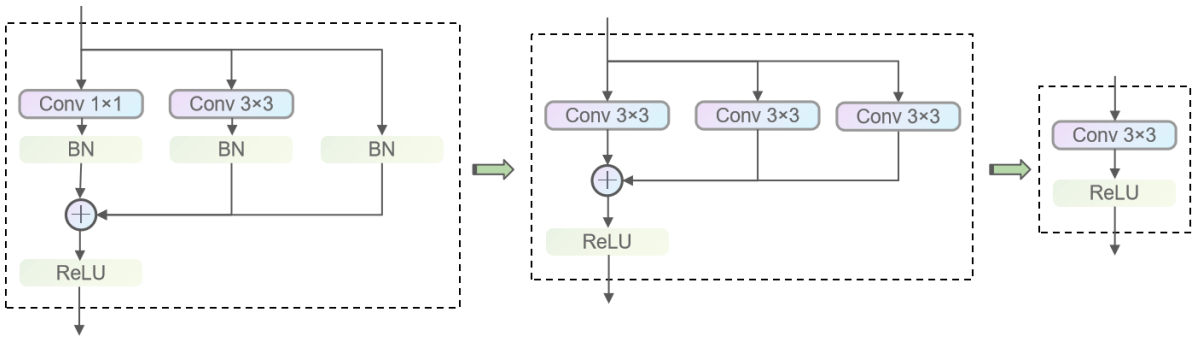


Figure 5. RepVGG structure reparameterization from training to inference.

Res2Net [38] replaces conventional 3×3 convolutions with multi-level residual connections to output multiscale features and increase the receptive field, as shown in Figure 6. Each level of 3×3 convolution takes the feature information output by the previous level as part of its input, increasing the receptive field for each convolution. As a result, Res2Net

obtains features with different sizes of receptive fields, enabling simultaneous extraction of detailed and global features. Experimental results [38] demonstrate significant performance improvements after integrating the Res2Net module into various models, verifying its effectiveness.

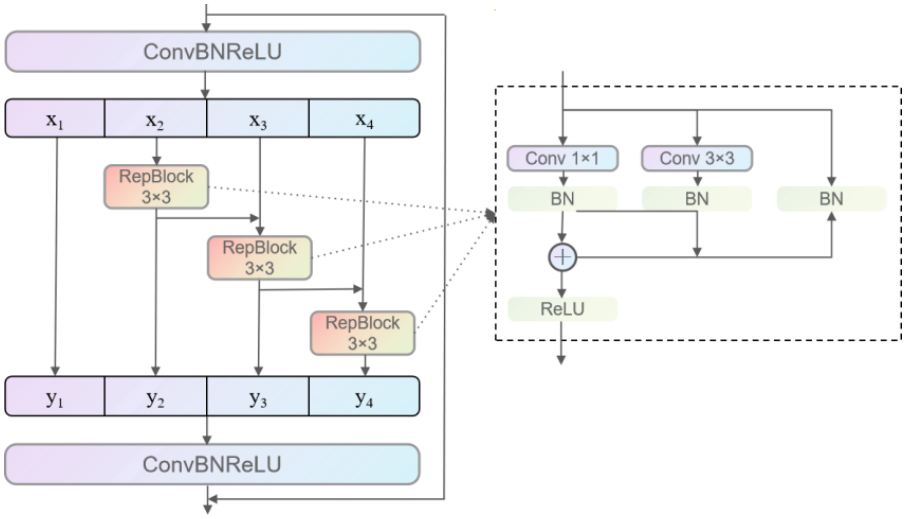


Figure 6. Architecture of Res2Rep.

Based on Res2Net and RepVGG, we propose a Res2Rep module, redesigning the 3×3 convolution structure in the Res2Net module through structural reparameterization, as shown in Figure 6. Incorporating the Res2Rep module into the foreground extraction network increases the effective receptive field, enhances the model's representational capability and foreground extraction accuracy, and fully leverages hardware computational power, significantly reducing inference latency.

3.2. Foreground Image Processing

3.2.1. Foreground Image Generation

Since the coverage of small and medium-sized objects in the detection results of the foreground extraction network is relatively low, the detected bounding boxes are expanded by a particular ratio to form new target regions, denoted as B_{coarse}. This approach aims to maximize the coverage of all objects in the original image. A greedy foreground region generation algorithm, as described in Algorithm 1, is proposed in this paper to merge the target regions.

Specifically, the expanded target region B_{coarse} is taken as input, and the first target region to be merged is selected as A. For each target region B in B_{coarse}, if there is an intersection between A and B, the boundary information of the merged region C is calculated. Then, the region information of A is updated to match that of C, and B is removed from the remaining target regions in B_{coarse}. This process is repeated until all target regions to be merged have been processed, resulting in a foreground sub-region denoted as A. The above process is iteratively performed until no target regions are left for merging.

The foreground region generation algorithm merges the detection results of the foreground extraction network into foreground aggregation regions. This paper adjusts the size of all foreground aggregation regions by scaling them by factors of $4\times$, $2\times$, $1\times$, and $0.7\times$ to control the size of the foreground image and improve the accuracy of small object recognition. The areas smaller than a threshold are enlarged, while those more significant than another are reduced (the thresholds are set to 32×32, 96×96, and 288×288, respectively).

Algorithm 1: Foreground Region Generation

Input: Bounding boxes B_{coarse}
Output: Merged regions B_{region}
1: **Initialize** The list of Merged regions B_{coarse}
2: **while** $B_{coarse} \neq \varnothing$ **do**
3: $B_{coarse} \leftarrow B_{coarse} - \{A\}$
4: **for all** $B \in B_{coarse}$ **do**
5: **if** $(|A| + |B|) \geq |C|$ **then**
6: $A = C$
7: $B_{coarse} \leftarrow B_{coarse} - \{B\}$
8: **end if**
9: **end for**
10: $B_{region} \leftarrow B_{region} \cup \{A\}$
11: **end while**
12: **return** B_{region}

Finally, the scaled foreground regions are concatenated into a foreground image using a greedy strategy. The foreground regions are sorted in descending order of their height. Based on the current available space, the foreground regions to be concatenated are divided into four priority levels: fully matching in height and width, slightly smaller width with matching height, slightly smaller height with matching width, and slightly smaller in both height and width. The priority is determined, and the foreground region with the highest priority is selected and concatenated into the available space. If multiple regions have the same priority, the one with the greatest height is chosen. This process is repeated until no foreground regions are left to be concatenated. The width of the blank space is adjusted based on the height of the concatenated result to generate the foreground image. The process is repeated until the foreground image with the smallest area is obtained.

3.2.2. Foreground Coordinate Mapping

The foreground image is used as input for the general detector to obtain the detection results of the foreground image. By mapping the position information of the sub-regions, the detection results can be transformed into the detection results of the original ultra-high-resolution image.

After concatenating the foreground image, the information of each foreground sub-region is represented as $r = [x_{reg11}, y_{reg11}, x_{reg22}, y_{reg22}, scale, x'_{reg11}, y'_{reg11}, x'_{reg22}, y'_{reg22}]$, where (x_{reg11}, y_{reg11}) and (x_{reg22}, y_{reg22}) are the coordinates of the upper-left and lower-right corners of the sub-region in the original ultra-high-resolution image, and (x'_{reg11}, y'_{reg11}) and (x'_{reg22}, y'_{reg22}) are the coordinates of the upper-left and lower-right corners of the sub-region in the foreground image, with *scale* representing the scaling factor of the foreground region. In the detection results of the foreground image, (x'_{obj11}, y'_{obj11}) and (x'_{obj22}, y'_{obj22}) are the coordinates of the upper-left and lower-right corners of the detection box in the foreground image.

We compute the intersection area between the bounding box and each foreground region to find the corresponding region for each bounding box. The region containing the bounding box is the region with the largest intersection area. To address cases where the bounding box boundary exceeds the boundary of the foreground region, we adjust the boundary to match the boundary of the corresponding foreground region. Subsequently, we employ Equation (1) to calculate the position of the bounding box in the ultra-high-resolution image.

$$\begin{cases} t_{obj11} = t_{reg11} + (t'_{obj11} - t'_{reg11}) \times scale \\ t_{obj22} = t_{reg22} - (t'_{obj22} - t'_{reg22}) \times scale \end{cases}, t \in \{x, y\} \quad (1)$$

4. Experimental Results and Analysis

4.1. Experimental Data

This study evaluates the effectiveness of the proposed model on the DOTA remote sensing dataset [12]. The DOTA dataset is currently the largest and most diverse dataset in aerial object detection, containing 2806 high-resolution aerial images. The images have a maximum resolution of 13,383 × 4287 and a minimum resolution of 475 × 547, with a median resolution of 4000 × 4000. The dataset is divided into validation, test, and training sets in a 1:2:3 ratio.

The dataset includes 15 common object categories with a total of 188,282 instances. The dataset is characterized by numerous small objects, as nearly all object sizes are within 5% of the image dimensions [12]. In alignment with convention [12], the object pixel size was evaluated using the height of the bounding boxes in HBB format. The statistical distribution of pixel values for specific object bounding boxes in the DOTA dataset is presented in Table 1.

Table 1. Object size distribution in the DOTA dataset.

Object Size	10–50 Pixel	50–300 Pixel	Above 300 Pixel
Ratio	0.57	0.41	0.02

The downscaling factor of the input image is estimated based on the general image resolution of 4000 × 4000. Assuming the input image is downscaled to 1024 × 1024 before input to the foreground extraction network, and the network performs three downsampling convolutions on the input image, the final two feature maps have sizes of 256 × 256 and 128 × 128, respectively. The 256 × 256 feature map is responsible for detecting small objects. At this point, the receptive field size of each feature map in the original ultra-high-resolution image is 15.6 × 15.6 pixels, meaning the network may have difficulty learning relevant features to detect objects with widths or heights smaller than 15.6 pixels. Based on the statistical results in Table 1, 15.6 pixels are sufficient to detect most objects, so this paper downsamples the ultra-high-resolution images in the DOTA dataset to 1024 × 1024 before inputting them to the foreground extraction network.

4.2. Implementation Details and Evaluation Metrics

We implement the foreground extraction network on Pytorch 1.10.2. All of our experiments use an NVIDIA RTX3090 GPU for training and testing. The training settings for the general detector are consistent with those in the original literature. In the experimental section of this paper, the training configurations for the general detectors remain consistent with the original literature. The foreground extraction network is trained based on a YOLOv5 model with a dimension of S [13]. Specific training settings entail a batch size of 8 and 300 epochs. Training employs the SGD optimizer, with an initial learning rate of 3×10^{-4} and a learning rate decay factor of 0.12.

The loss curve is illustrated in Figure 7. Within YOLOv5, three loss functions are employed: classification loss, bounding box localization loss, and object confidence loss. The localization loss quantifies the positional error between predicted and annotated boxes, the classification loss assesses the precision of predicted class assignments, and the confidence loss determines the presence of objects. Given that the foreground extraction network eliminates the target classification component of YOLOv5, we depict the curves for localization loss and confidence loss while omitting the classification loss curve.

To evaluate the effectiveness of our proposed method, we select the following evaluation metrics: parameter (parameter count), FLOPs (floating-point operations), mAP (mean average precision), FPS (frames per second), FR (foreground ratio), FOR (foreground object number ratio), and FAR (foreground area ratio).

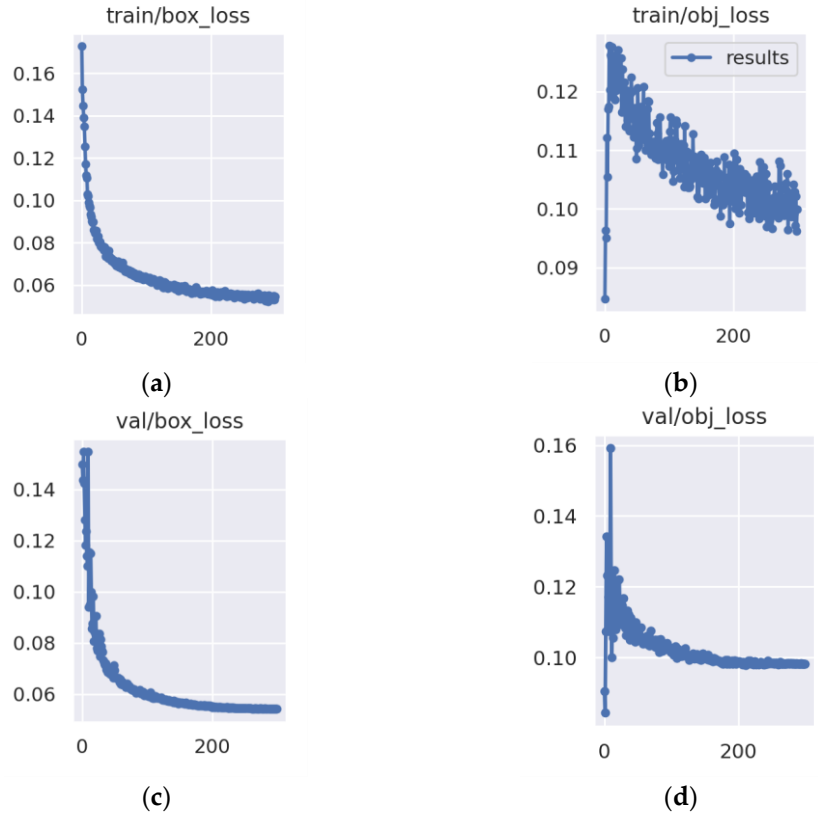


Figure 7. Visualization of training and validation loss curves for foreground extraction network. (**a**) Training localization loss; (**b**) training confidence loss; (**c**) validation localization loss; (**d**) validation confidence loss.

FLOPs quantify the floating-point operations the model requires, providing a measure of model complexity. FPS denotes the number of images processed per second, serving as a pivotal indicator of processing speed. Mean average precision (mAP) is the mean value of the average precisions (AP) across all classes, which measures the detection performance of the model. AP, in turn, is the area under the precision–recall curve. Both AP and mAP are computed using Equations (2) and (3), where c represents the number of classes in the dataset, and P and R denote precision and recall, respectively.

$$AP = \int_0^1 P(R)dR \tag{2}$$

$$mAP = \frac{1}{c}\sum_{i=1}^{c} AP_i \tag{3}$$

The foreground ratio describes the proportion of foreground object area to the total area of the image, the foreground object ratio describes the ratio of the number of objects in the foreground image to the number of objects in the original image, and the foreground area ratio describes the ratio of the area of the foreground image to the area of the original image.

4.3. Experiment on Bounding Box Scaling Factor

Foreground extraction is a pivotal step in our object detection methodology, where the selection of different scaling factors can notably impact the generation of foreground images, consequently influencing object detection performance. In our investigation, we conducted comprehensive experiments on varying scaling factors, analyzing the changes in FR, FOR, and FAR, as presented in Table 2.

Table 2. Experiment on the bounding box scaling factor.

Scaling Factor	Foreground Image Size [1]	FR	FOR	FAR
1.1	777 × 817	0.20	0.44	0.33
1.3	913 × 963	0.21	0.66	0.47
1.35	942 × 1001	0.21	0.68	0.48
1.4	971 × 1038	0.20	0.70	0.52
1.5	1024 × 1107	0.19	0.72	0.58
1.7	1091 × 1204	0.16	0.75	0.73

[1] The width and height of foreground images were computed as the average of all individual foreground image widths and heights, respectively.

Based on our experimental findings, we discerned increased detection accuracy and number of target objects within the foreground image as the scaling factor augmented. When the scaling factor reached 1.35, the foreground ratio achieved its peak value, and the count of target objects in the foreground image also approached its maximum. In contrast, the relative area ratio of the foreground region remained moderate. This implies that we could effectively extract foreground targets from the image while maintaining a relatively high detection accuracy. Conversely, as the scaling factor exceeded 1.35, the area of the foreground region surged dramatically. However, the increase in the count of target objects within the foreground image remained limited, resulting in a decrease in the foreground ratio. Considering that the objective of foreground extraction is to encompass more objects while maintaining a smaller image size, our experimental results advocate for 1.35 as a balance point that can effectively enhance the foreground ratio. By judiciously considering both target detection accuracy and foreground ratio, the utilization of 1.35 as the boundary box scaling factor yields optimal performance for our approach in addressing the foreground extraction task in ultra-high-resolution images.

4.4. Ablation Experiment

To evaluate the impact of each component in the foreground extraction network on foreground region extraction, we use the YOLOv5 [13] model with a size of S from version 6.0 as the baseline model and gradually add components to demonstrate their effectiveness. The results are shown in Table 3.

Table 3. Ablation experiment of the foreground extraction network.

w/o PH	w/o Classifier	C3Faster	Res2Rep	Parameter	FLOPs	Model Size	FR	FOR	FAR	FPS
				7,235,389	16.5 G	13.8 M	0.22	0.64	0.44	172
✔ [1]	✔			604,600	12.8 G	1.78 M	0.21	0.62	0.44	188
✔	✔	✔		509,944	10.5 G	1.60 M	0.22	0.61	0.42	196
✔	✔	✔	✔	879,032	20.6 G	2.73 M	0.21	0.68	0.48	107

[1] "✔" denotes the method or module indicated in the table's header is used.

The ablation experiment yields the following conclusions: with only 12.1% of the parameter count of the YOLOv5 baseline model and a 24.8% increase in floating-point calculations, the foreground image extraction method decreases the foreground ratio by 1% and increases the foreground object ratio and foreground area ratio by 4%.

Removing the large object detection head and classifier reduces the model parameter count by 91.6%. After incorporating the C3Faster module, the parameter count is reduced by 15.7%, resulting in improved foreground extraction speed and a 1% decrease in extracted foreground objects. The addition of the Res2Rep module, despite increasing floating-point calculations and decreasing FPS, significantly enhances foreground extraction performance compared to the baseline.

The experimental results demonstrate the following:

1. Removing the large object detection head and classifier has little effect on foreground extraction while effectively reducing the model size and computational complexity, thus enhancing foreground extraction efficiency;
2. The C3Faster module further accelerates foreground extraction speed;
3. The Res2Rep module effectively improves foreground extraction performance.

4.5. Comparison Experiment

To verify the processing speed of the foreground extraction method for ultra-high-resolution images, we conducted comparative experiments on the DOTA dataset using sliding window cropping to different sizes and the foreground extraction method, as shown in Table 4.

Table 4. Comparison of processing results and processing time for ultra-high-resolution images using sliding window cropping and foreground image methods.

Method	Average Resolution		Overlap	Total	Average Image Count	Mean Speed
	Original Image	Result				
Sliding Window Cropping	2398 × 2210	2048 × 2048	200	6380	3.41	0.58 s
		1024 × 1024	200	21,046	11.26	0.51 s
		600 × 600	150	65,644	35.12	0.49 s
Foreground Image		942 × 1001	——	1865	1	0.12 s

When cropping the DOTA dataset using the sliding window method, two cropping settings are employed: the image is cropped into 1024 × 1024 patches with an overlap of 200 pixels or into 600 × 600 patches with an overlap of 150 pixels, as described in references [2–5,9]. Additionally, although the general detection network cannot handle images of size 2048 × 2048, the original images are also cropped into 2048 × 2048 patches for comparison purposes.

After foreground extraction on the DOTA dataset, the average resolution of the foreground image is 942 × 1001, slightly smaller than the commonly used cropping size of 1024 × 1024. However, cropping the image into 1024 × 1024 patches would result in 21,046 images, with an average of 11.26 patches per original image, while foreground extraction generates only one foreground image for each original image. Consequently, the sliding window cropping method requires the model to detect 11.26 images for each original image, while the foreground image method only needs to detect one image. Even when the cropping size is set to 2048 × 2048, the average number of detection images is 3.41 times higher than that of the foreground image extraction method.

To evaluate the detection performance of the foreground image, Table 5 compares detection results using different detection algorithms for both foreground and cropped images.

Table 5. Comparison experiment of different detection algorithms.

Method	Image Type	mAP	FPS
Two-stage			
RoI Trans. [39]	CI [1]	69.6	18
	FI [2]	64.6	
Gilding Vertex [40]	CI	75.0	21
	FI	66.2	
Oriented R-CNN [41]	CI	75.9	20
	FI	66.3	
QPDet [2]	CI	76.3	21
	FI	66.7	

Table 5. *Cont.*

Method	Image Type	mAP	FPS
One-stage			
S2ANet [42]	CI	74.1	20
	FI	66.5	
MDCT [43]	CI	75.7	13
	FI	66.8	
G-Rep [3]	CI	75.6	19
	FI	66.2	
CoF-Net [44]	CI	77.2	18
	FI	66.7	

[1] " CI " indicates use of cropped image detection. [2] " FI " indicates use of foreground image detection.

The state-of-the-art detection algorithm CoF-Net is used as the general detector to evaluate the detection performance of the foreground image extraction method and the sliding window cropping method. Cropping the original DOTA images into 1024 × 1024 patches and detecting each patch takes 1.14 s, while the proposed foreground image extraction method takes only 0.18 s to detect each original image. Therefore, using the foreground image extraction method reduces the detection time per original image by 83.8% compared to the sliding window cropping method with a 1024 × 1024 patch.

The experimental results show that on the DOTA dataset with ultra-high-resolution images, the proposed foreground image method achieves an average mAP of 8.7% lower than the cropping method with patches. However, the average detection time is reduced by 83.8%. This validates the effectiveness of the proposed method for fast object detection in ultra-high-resolution images. Due to the low foreground ratio and object aggregation characteristics of ultra-high-resolution images, many patches in the cropping method do not contain any objects, resulting in a waste of resources in background image object detection. The foreground image method significantly increases the foreground ratio and is more efficient in object detection, making it more suitable for fast object detection in ultra-high-resolution images.

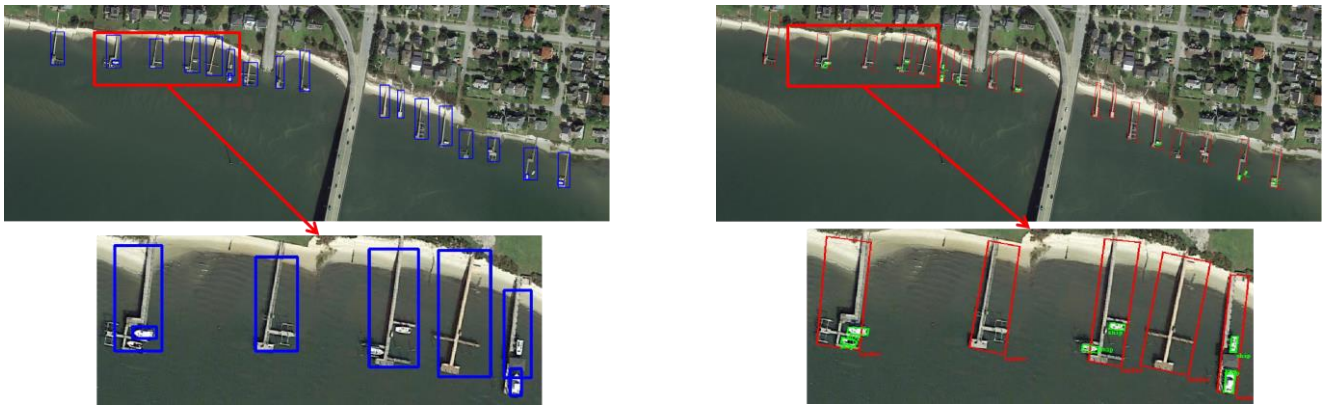
4.6. Visualization Analysis

Some foreground image results are visualized in Figure 8. To analyze the impact of the foreground map approach on object detection performance, we selected two types of sample images from the DOTA test set: one with small objects close to large objects and the other with small objects far from large objects. We used the current state-of-the-art rotation-based object detector, CoF-Net, as the baseline detector for comparison. The detection results are shown in Figures 9 and 10.

The foreground map approach involves scaling and merging the extracted target regions to obtain the foreground region, which is then concatenated to form the foreground map. During the extraction of the foreground region, some sparsely distributed small object regions may be omitted, making it challenging for the foreground map approach to detect targets in these sparse regions. On the other hand, since the foreground map approach enlarges the small object clusters during the generation of the foreground map, it is more likely to identify targets in regions with clustered small objects than the standard method.

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

Figure 8. Visualization of foreground images. (**a**) Original image; (**b**) foreground extraction network's recognition result; (**c**) foreground image generated from the original image.



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

Figure 9. Detection results of images with small objects near large objects. (**a**) Foreground extraction network's recognition result; (**b**) final recognition result.

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

Figure 10. Detection results of images with small objects far from large objects. (**a**) Foreground extraction network's recognition result; (**b**) final recognition result.

5. Conclusions

In this study, we introduced a novel object detection framework named MegaDetectNet, tailored specifically to object detection in ultra-high-resolution images. With a thorough consideration of the characteristics of foreground extraction and building upon the foundation of YOLOv5, we devised an efficient foreground extraction network to delineate foreground regions within ultra-high-resolution images. Leveraging a greedy foreground region fusion and concatenation algorithm, we successfully generated foreground images which supplanted the original ultra-high-resolution images as input for object detection, resulting in a marked enhancement in detection efficiency.

Our experimental findings unequivocally underscore the superiority of the MegaDetectNet approach. On the DOTA dataset, compared to conventional sub-image splitting methods, our approach demonstrated an average time saving of 83.8%. This outcome not only underscores the practicality of the MegaDetectNet approach in the realm of object detection for ultra-high-resolution images but also validates the efficacy of our foreground extraction strategy.

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

Funding: This research was funded by the National Innovation Special Zone Project, grant number 2016300TS00600113.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available upon request from the corresponding author.

Acknowledgments: The authors would like to acknowledge the anonymous reviewers and editors whose thoughtful comments helped to improve this manuscript.

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

References

1. Van Etten, A. You only look twice: Rapid multi-scale object detection in satellite imagery. *arXiv* **2018**, arXiv:1805.09512.
2. Yao, Y.; Cheng, G.; Wang, G.; Li, S.; Zhou, P.; Xie, X.; Han, J. On improving bounding box representations for oriented object detection. *IEEE Trans. Geosci. Remote Sens.* **2022**, *61*, 5600111. [CrossRef]
3. Hou, L.; Lu, K.; Yang, X.; Li, Y.; Xue, J. G-rep: Gaussian representation for arbitrary-oriented object detection. *Remote Sens.* **2023**, *15*, 757. [CrossRef]

4. Wang, D.; Zhang, Q.; Xu, Y.; Zhang, J.; Du, B.; Tao, D.; Zhang, L. Advancing plain vision transformer toward remote sensing foundation model. *IEEE Trans. Geosci. Remote Sens.* **2022**, *61*, 5607315. [CrossRef]
5. Wu, Y.; Li, J. YOLOv4 with Deformable-Embedding-Transformer Feature Extractor for Exact Object Detection in Aerial Imagery. *Sensors* **2023**, *23*, 2522. [CrossRef]
6. Li, Y.; Wu, C.; Fan, H.; Mangalam, K.; Xiong, B.; Malik, J.; Feichtenhofer, C. Improved multiscale vision transformers for classification and detection. *arXiv* **2021**, arXiv:2112.01526.
7. Wang, L.; Liu, X.; Ma, J.; Su, W.; Li, H. Real-Time Steel Surface Defect Detection with Improved Multi-Scale YOLO-v5. *Processes* **2023**, *11*, 1357. [CrossRef]
8. Ying, X.; Wang, Q.; Li, X.; Yu, M.; Jiang, H.; Gao, J.; Liu, Z.; Yu, R. Multi-attention object detection model in remote sensing images based on multi-scale. *IEEE Access* **2019**, *7*, 94508–94519. [CrossRef]
9. Yang, X.; Yan, J.; Liao, W.; Yang, X.; Tang, J.; He, T. Scrdet++: Detecting small, cluttered and rotated objects via instance-level feature denoising and rotation loss smoothing. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *45*, 2384–2399. [CrossRef]
10. Zhao, Q.; Liu, B.; Lyu, S.; Wang, C.; Zhang, H. TPH-YOLOv5++: Boosting Object Detection on Drone-Captured Scenarios with Cross-Layer Asymmetric Transformer. *Remote Sens.* **2023**, *15*, 1687. [CrossRef]
11. Zhang, Q.; Xu, Y.; Zhang, J.; Tao, D. Vsa: Learning varied-size window attention in vision transformers. In Proceedings of the European Conference on Computer Vision, Tel Aviv, Israel, 23–27 October 2022; pp. 466–483.
12. Ding, J.; Xue, N.; Xia, G.-S.; Bai, X.; Yang, W.; Yang, M.Y.; Belongie, S.; Luo, J.; Datcu, M.; Pelillo, M. Object detection in aerial images: A large-scale benchmark and challenges. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *44*, 7778–7796. [CrossRef] [PubMed]
13. Jocher, G.; Stoken, A.; Chaurasia, A.; Borovec, J.; Kwon, Y.; Michael, K.; Changyu, L.; Fang, J.; Skalski, P.; Hogan, A. *Ultralytics/yolov5: v6.0-YOLOv5n'NANO'MODELS, Roboflow Integration, TensorFlow Export, OpenCV DNN Support*; Zenodo: Honolulu, HI, USA, 2021.
14. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *Adv. Neural Inf. Process. Syst.* **2015**, *39*, 91–99. [CrossRef] [PubMed]
15. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2961–2969.
16. Bochkovskiy, A.; Wang, C.-Y.; Liao, H.-Y.M. Yolov4: Optimal speed and accuracy of object detection. *arXiv* **2020**, arXiv:2004.10934.
17. Wang, C.-Y.; Bochkovskiy, A.; Liao, H.-Y.M. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 18–22 June 2023; pp. 7464–7475.
18. Lin, T.-Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal loss for dense object detection. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2980–2988.
19. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.-Y.; Berg, A.C. Ssd: Single shot multibox detector. In Proceedings of the European Conference on Computer Vision (ECCV), Amsterdam, The Netherlands, 8–16 October 2016; pp. 21–37.
20. Tian, Z.; Shen, C.; Chen, H.; He, T. Fcos: Fully convolutional one-stage object detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 9627–9636.
21. 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.
22. Li, C.; Yang, T.; Zhu, S.; Chen, C.; Guan, S. Density map guided object detection in aerial images. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, Seattle, WA, USA, 14–19 June 2020; pp. 190–191.
23. Wang, Y.; Yang, Y.; Zhao, X. Object detection using clustering algorithm adaptive searching regions in aerial images. Proceedings of European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; pp. 651–664.
24. 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]
25. 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 2022, Virtual, 22 February–1 March 2022; pp. 1026–1033.
26. Singh, B.; Davis, L.S. An analysis of scale invariance in object detection snip. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 3578–3587.
27. Liu, S.; Qi, L.; Qin, H.; Shi, J.; Jia, J. Path aggregation network for instance segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 8759–8768.
28. Qin, Z.; Li, Z.; Zhang, Z.; Bao, Y.; Yu, G.; Peng, Y.; Sun, J. ThunderNet: Towards real-time generic object detection on mobile devices. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 6718–6727.
29. Jiang, Y.; Tan, Z.; Wang, J.; Sun, X.; Lin, M.; Li, H. GiraffeDet: A heavy-neck paradigm for object detection. *arXiv* **2022**, arXiv:2202.04256.
30. Yang, C.; Huang, Z.; Wang, N. QueryDet: Cascaded sparse query for accelerating high-resolution small object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 13668–13677.
31. He, L.; Todorovic, S. DESTR: Object detection with split transformer. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 9377–9386.

32. Zhu, H.; Ke, W.; Li, D.; Liu, J.; Tian, L.; Shan, Y. Dual cross-attention learning for fine-grained visual categorization and object re-identification. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 4692–4702.
33. Chen, J.; Kao, S.-h.; He, H.; Zhuo, W.; Wen, S.; Lee, C.-H.; Chan, S.-H.G. Run, Don't Walk: Chasing Higher FLOPS for Faster Neural Networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 18–22 June 2023; pp. 12021–12031.
34. Han, K.; Wang, Y.; Tian, Q.; Guo, J.; Xu, C.; Xu, C. Ghostnet: More features from cheap operations. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 1580–1589.
35. Zhang, Q.; Jiang, Z.; Lu, Q.; Han, J.n.; Zeng, Z.; Gao, S.-H.; Men, A. Split to be slim: An overlooked redundancy in vanilla convolution. *arXiv* **2020**, arXiv:2006.12085.
36. Ding, X.; Zhang, X.; Ma, N.; Han, J.; Ding, G.; Sun, J. Repvgg: Making vgg-style convnets great again. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Virtual, 19–25 June 2021; pp. 13733–13742.
37. Li, C.; Li, L.; Jiang, H.; Weng, K.; Geng, Y.; Li, L.; Ke, Z.; Li, Q.; Cheng, M.; Nie, W. YOLOv6: A single-stage object detection framework for industrial applications. *arXiv* **2022**, arXiv:2209.02976.
38. Gao, S.-H.; Cheng, M.-M.; Zhao, K.; Zhang, X.-Y.; Yang, M.-H.; Torr, P. Res2net: A new multi-scale backbone architecture. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**, *43*, 652–662. [CrossRef]
39. Ding, J.; Xue, N.; Long, Y.; Xia, G.-S.; Lu, Q. Learning RoI transformer for oriented object detection in aerial images. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 2849–2858.
40. Xu, Y.; Fu, M.; Wang, Q.; Wang, Y.; Chen, K.; Xia, G.-S.; Bai, X. Gliding vertex on the horizontal bounding box for multi-oriented object detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *43*, 1452–1459. [CrossRef]
41. Xie, X.; Cheng, G.; Wang, J.; Yao, X.; Han, J. Oriented R-CNN for object detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Virtual, 11–17 October 2021; pp. 3520–3529.
42. Han, J.; Ding, J.; Li, J.; Xia, G.-S. Align deep features for oriented object detection. *IEEE Trans. Geosci. Remote Sens.* **2021**, *60*, 5602511. [CrossRef]
43. Chen, J.; Hong, H.; Song, B.; Guo, J.; Chen, C.; Xu, J. MDCT: Multi-Kernel Dilated Convolution and Transformer for One-Stage Object Detection of Remote Sensing Images. *Remote Sens.* **2023**, *15*, 371. [CrossRef]
44. Zhang, C.; Lam, K.-M.; Wang, Q. Cof-net: A progressive coarse-to-fine framework for object detection in remote-sensing imagery. *IEEE Trans. Geosci. Remote Sens.* **2023**, *61*, 5600617. [CrossRef]