remote sensing

MDPI

Article

A Lightweight Remote Sensing Aircraft Object Detection Network Based on Improved YOLOv5n

Jiale Wang [1,2], Zhe Bai [1], Ximing Zhang [1,*] and Yuehong Qiu [1]

[1] Xi'an Institute of Optics and Precision Mechanics of CAS, Xi'an 710119, China; wangjiale211@mails.ucas.ac.cn (J.W.); byf@opt.ac.cn (Z.B.); yhqiu@opt.ac.cn (Y.Q.)
[2] University of Chinese Academy of Sciences, Beijing 100049, China
* Correspondence: zhangximing@opt.ac.cn

Abstract: Due to the issues of remote sensing object detection algorithms based on deep learning, such as a high number of network parameters, large model size, and high computational requirements, it is challenging to deploy them on small mobile devices. This paper proposes an extremely lightweight remote sensing aircraft object detection network based on the improved YOLOv5n. This network combines Shufflenet v2 and YOLOv5n, significantly reducing the network size while ensuring high detection accuracy. It substitutes the original CIoU and convolution with EIoU and deformable convolution, optimizing for the small-scale characteristics of aircraft objects and further accelerating convergence and improving regression accuracy. Additionally, a coordinate attention (CA) mechanism is introduced at the end of the backbone to focus on orientation perception and positional information. We conducted a series of experiments, comparing our method with networks like GhostNet, PP-LCNet, MobileNetV3, and MobileNetV3s, and performed detailed ablation studies. The experimental results on the Mar20 public dataset indicate that, compared to the original YOLOv5n network, our lightweight network has only about one-fifth of its parameter count, with only a slight decrease of 2.7% in mAP@0.5. At the same time, compared with other lightweight networks of the same magnitude, our network achieves an effective balance between detection accuracy and resource consumption such as memory and computing power, providing a novel solution for the implementation and hardware deployment of lightweight remote sensing object detection networks.

Keywords: deep learning; lightweight network; YOLOv5n; Shufflenet v2; CA; EIoU loss; deformable convolution

Citation: Wang, J.; Bai, Z.; Zhang, X.; Qiu, Y. A Lightweight Remote Sensing Aircraft Object Detection Network Based on Improved YOLOv5n. *Remote Sens.* **2024**, *16*, 857. https://doi.org/10.3390/rs16050857

Academic Editor: Pedro Melo-Pinto

Received: 15 January 2024
Revised: 26 February 2024
Accepted: 26 February 2024
Published: 29 February 2024

1. Introduction

Remote sensing aircraft object detection is crucial in various applications. In civil aviation and the aerospace industry, it helps identify other aircraft, drones, or obstacles around an aircraft to prevent collisions and enhance aviation safety. It also aids in the real-time monitoring and tracking of civil aviation flights, cargo planes, and private aircrafts to ensure their flight path and status. In the military domain, it identifies and tracks enemy aircraft, performs aerial reconnaissance, gathers intelligence, and supports aerial strikes and combat. In emergencies like aircraft disappearance or deviation from course, it assists in search and rescue operations to locate the aircraft and passengers.

The breadth of application domains in remote sensing object detection corresponds to the complexity of target data sources. Particularly for remote sensing targets, completing object detection tasks depends not only on the algorithm but also on the data source. Factors affecting detection effectiveness include not only the reliability of artificial intelligence algorithms but also that of remote sensing data sources. From acquiring raw data to inputting them into the network, considerations extend beyond complex scenes to factors like imaging conditions, resolution, and storage processes. During imaging, sensor size, atmospheric conditions, observation time, and lighting affect results. For instance, adverse

weather like rain and fog significantly degrade imaging quality, necessitating image dehazing during processing. Additionally, image denoising is crucial in preprocessing due to signal transmission, dark current, and random noise effects.

For algorithms, considerations during training involve sample quantity, scene types, target categories, and annotation quality to select high-quality datasets. Enhancing recognition involves designing new network structures, adjusting training strategies, and selecting parameters. Since algorithms often deploy on hardware with low computing power and memory, compressing network volume and parameter size is vital. Given these considerations, selecting a suitable algorithm framework and dataset is paramount.

Currently, remote sensing object detection faces several challenges. Firstly, it suffers from a single detection perspective, limiting the useful information gathered from an overhead view. Secondly, remote sensing targets tend to be small in size with minimal differences between types, making fine-grained identification challenging. Most significantly, the vast range of object detection on remote sensing images leads to substantial computing power consumption and inefficient detection processes. To address these challenges, various algorithms with exceptional performance have emerged in remote sensing object detection, such as the R-CNN series [1–3], YOLO series [4–8], and DETR [9]. Nonetheless, deploying these algorithms on mobile hardware platforms with constrained computing and memory resources presents significant challenges due to their large network size and complex structures. In high-altitude environments where real-time imaging and object detection are crucial, available memory and computational resources are severely limited. Addressing actual memory usage, after training these algorithms on the same dataset, the resulting model files typically range from a few megabytes to hundreds of megabytes. Some exceed even hundreds of megabytes, which is unacceptable to a certain extent. From a computational perspective, existing remote sensing aircraft object detection networks prioritize performance metrics, often leading to increased network depth and width. While this may improve performance, it also substantially increases resource consumption in terms of memory and computation. For instance, the actual FLOPs of YOLOv5n can reach as high as 4.3 G, rendering existing algorithms impractical for deployment on some actual mobile hardware platforms.

In response, several lightweight object detection networks have been proposed. Examples include the MobileNet series [10–12], Ghost-Net series [13], PP-LCNet series [14], and Shufflenet series [15,16]. However, each of these algorithms has shortcomings in terms of detection performance, network size, and resource consumption. They fail to strike a balance between detection performance and memory and computational resource utilization. For example, Ghost-Net and Mobile-Net have excessively large network parameter sizes, while PP-LCNet exhibits slightly inferior detection performance. This makes it challenging to meet practical application requirements. Therefore, while ensuring detection performance remains unchanged or slightly decreased, the focus should be on significantly reducing network parameters and computational load to achieve a perfect balance between detection performance and resource consumption.

To achieve real-time remote sensing object detection while ensuring the network remains lightweight, we chose YOLOv5n as our baseline network. The You Only Look Once (YOLO) series [4–8] of algorithms is a prominent representative of object detection algorithms, with YOLOv5 being one of the most mature algorithms in this series, striking a good balance between speed and accuracy. YOLOv5n, being the smallest model in the series, has optimized detection speed to the fullest. However, despite these advancements, the increasing real-time demands of remote sensing object detection systems pose challenges due to the complexity and size of network models and parameters, making it challenging to run on hardware with low power consumption and computing power. Hence, lightweighting the network becomes imperative.

For lightweight processing of the YOLO series, two main technical approaches are prevalent. One involves replacing the backbone network with a lightweight alternative, while the other entails adjusting the convolution structure, such as modifying the size

of convolution kernels and the total number of kernels, which determine the width of the network.

In our pursuit of maintaining detection accuracy akin to the original network, we conducted numerous hypotheses and experiments aimed at minimizing the volumetric dimensions of the network model. The Backbone, a critical and parameter-intensive component of YOLO series algorithms, is primarily responsible for extracting features from input remote sensing images and conveying them to the Neck. To enhance cost-effectiveness in terms of both parameter quantity and performance, we drew inspiration from the Shufflenet v2 paradigm. Utilizing group convolution and channel shuffling principles, we adopted the ShuffleUnit as the fundamental unit in constructing the feature extraction network. The ShuffleUnit facilitates deep convolution operations, achieving group convolution through the connection of two branches and channel reordering, thereby enhancing the network's non-linear expressive capability while reducing computational costs. Additionally, we introduced the forward module to implement forward propagation, involving channel splitting, branch computation, and channel reordering. Unlike many similar lightweight networks, we departed from the alternating connection pattern of the C3 module and lightweight module in the original YOLOv5 network. Instead, we arranged six ShuffleUnits in a serial fashion to achieve the utmost lightweight design. To further enhance network performance, we embedded the Coordinate Attention (CA) mechanism at the end of the backbone. Moreover, we replaced the original Complete Intersection over Union (CIoU) loss function with the more advanced Efficient Intersection over Union (EIoU). Finally, to better handle deformations in irregular regions and targets in remote sensing images, we optimized a portion of traditional convolutions into deformable convolutions, resulting in a reduction in the overall parameter count.

The subsequent sections of this paper are organized as follows: Section 1 provides an extensive overview of remote sensing object detection algorithms and attention mechanisms, particularly organizing algorithms based on YOLOv5. Section 2 outlines the overall framework of our proposed network and elaborates on the improvements made. In Section 3, we utilize YOLOv5 as the baseline network and employ various popular lightweight networks as replacements for the backbone, conducting comprehensive experimental comparisons across detection accuracy, parameter count, model size, and floating-point operations. These experiments convincingly demonstrate the superiority of our network. Furthermore, Section 4 includes ablation experiments to validate the feasibility of our design modules. Finally, in Section 5, we offer a comprehensive summary of the research and articulate future prospects.

2. Related Works

2.1. Remote Sensing Object Detection Based on Deep Learning

Around 2014, deep learning-based approaches began to dominate the field of object detection, subsequently extending their influence to the entire domain of remote sensing object detection. Object detection algorithms based on deep learning are typically categorized into two types: single-stage object detection algorithms and two-stage object detection algorithms. [17] Two-stage algorithms mainly include Region-CNN (R-CNN) [1], Fast R-CNN [2], Faster R-CNN [3], and others. They typically start by using a Region Proposal Network (RPN) [3] to generate candidate boxes based on the texture, color, and size features associated with the objects. These candidate boxes undergo a filtering process to reduce their number before being sent to a deep learning network, typically utilizing a Convolutional Neural Network (CNN) [18] for feature extraction. The obtained feature vectors are compared with predefined target categories to confirm the presence of a target and perform target classification. Simultaneously, each candidate box undergoes position regression to obtain corresponding position coordinate information. Due to its excellent detection speed and relatively small resource consumption, single-stage networks have gained dominance in recent years. The most prevalent network among single-stage networks is the YOLO series algorithm, known for its fast detection speed and high accuracy,

particularly suitable for real-time scenarios. The single-stage algorithm mainly includes the YOLO series and the Single Shot MultiBox Detector (SSD) [19] series. They generally pass the entire image as input to the CNN, skipping the step of generating candidate boxes. The CNN directly outputs category and location information of the target. Single-stage networks typically use-predefined anchor boxes with different scale sizes and aspect ratios to process targets, judging the presence of a target in each anchor box and predicting its location and category for object detection. With ongoing research, the detection accuracy of single-stage networks continues to improve. With superior detection speed, they have gradually replaced two-stage algorithms in many engineering practices, becoming the mainstream in practical applications.

Traditional methods for object detection in remote sensing images have limited representation power. Recently, many deep learning-based networks specifically for remote sensing images have emerged. Reference [20] proposed a method that redesigns the feature extractor, utilizes a multi-scale object proposal network (MS-OPN) for object-like region generation, and employs an accurate object detection network (AODN) for object detection based on fused feature maps. [21] Reference [22] introduced the CSand-Glass module to replace the residual module in the backbone feature extraction network of YOLOv5, achieving higher accuracy and speed in remote sensing images. Liu et al. proposed the YOLO-extract algorithm [23], which optimizes the model structure of YOLOv5 in two main ways. Firstly, it integrates a new feature extractor with stronger feature extraction ability. Secondly, it incorporates Coordinate Attention into the network.

In recent years, the Transformer [24] has had a profound impact on deep learning. In the field of object detection, Facebook introduced the end-to-end object detection network called DEtection Transformer (DETR) [9], which is based on the Transformer architecture. DETR can be viewed as a transformation process from an image sequence to a set sequence. This is due to the inherent nature of the Transformer as a sequence-to-sequence transformer. The approach taken by DETR involves unfolding the pixels of the output feature map from the backbone into a one-dimensional sequence, treating it as the sequence length, while maintaining the definitions of batch and channel. Consequently, DETR is capable of computing the correlations between each pixel of the feature map and all other pixels, unlike in CNNs, where this is achieved through the receptive field. The Transformer demonstrates the ability to capture a larger perceptual range than CNNs.

In addition to the previously mentioned methods for remote sensing object detection, there is a particular significance in introducing arbitrary-oriented remote sensing object detection methods. From a training perspective, the primary distinction between arbitrary-oriented object detection and regular object detection is in how object bounding boxes are represented in the dataset. Convolutional neural networks struggle to capture variations in scale and orientation of objects in remote sensing images. This struggle arises from the limited generalization ability of convolutional operations to target rotation and scale changes. As a result, the detection performance of convolutional neural networks tends to decrease, especially when dealing with dense objects and remote sensing targets with centrally symmetric features.

Optimizing the loss function for representing object bounding boxes is currently a focus of research. The early work of DRBox [25] has significantly advanced arbitrary-oriented remote sensing object detection. DRBox identified various challenges in this field and introduced three different models along with their parameters tailored for cars, ships, and aircraft, encompassing a wide range of differently sized targets and distinguishing their heads and tails. R3det [26] adopts single-stage object detection and suggests re-encoding the position information of refined bounding boxes into corresponding feature points. This process reconstructs the entire feature map to achieve feature alignment. The ROI-Transformer [27] introduces a module named RoI Transformer, which detects directed and dense objects using supervised RRoI learning and position-sensitive alignment-based feature extraction within a two-stage framework. The innovative application of polar

coordinates in the P-RSDet [28] reduces parameter volume and introduces a novel loss function, Polar Ring Area Loss, leading to enhanced detection performance.

2.2. Lightweight Methods for Object Detection Networks Based on Deep Learning

Deep learning methods have demonstrated significant advancements in remote sensing object detection in recent years. As previously mentioned, the wide scope of object detection in remote sensing images leads to substantial computational overhead and significantly lowers detector efficiency. Consequently, lightweighting network models has become a focal point in current research. The primary goal of lightweighting network models is to reduce model complexity and decrease the number of model parameters. Four primary technical approaches exist for lightweighting network models: compressing pretrained large models, redesigning lightweight models, accelerating numerical operations, and hardware acceleration.

In current practice, the first three technical approaches are widely employed. Knowledge distillation and model pruning are both well-established methods for compressing models. Knowledge distillation, a common method for model compression, reduces model volume and parameter count by transferring the knowledge of a complex model to a lightweight one. The literature points out that this method extracts the knowledge contained in the complex "teacher" model that has been trained into another lightweight model, the "student" model. In addition, model pruning is also a common method. Model pruning reduces model size by removing unimportant weights or neurons. It can be categorized into structured pruning and unstructured pruning based on different methods. Furthermore, adjusting the number and size of the network's convolution kernels can achieve model lightweighting. Generally, a larger convolution kernel size can enhance feature extraction. However, the literature indicates that large convolution kernels increase computational requirements and the number of parameters, leading to unstable gradients. Therefore, scholars often employ multiple small convolution kernels instead of a single large one to compress models, as seen in the Visual Geometry Group (VGG) network proposed in [29].

Lightweight networks find extensive applications in remote sensing image processing. The design of lightweight network architecture originated with SqueezeNet [30] in 2016 and MobileNet [10–12] in 2017. Subsequently, new and improved networks, such as SqueezeNext [31] and MobileNetV2 [11], have emerged. MobileNetV1 can essentially be viewed as replacing the standard convolutional layer in VGG with depthwise separable convolution [32]. MobileNetV2 [11] introduces shortcut connections and replaces part of ReLU with a linear activation function. They employ pointwise convolution to increase the dimension prior to depth convolution, extract features through depth convolution, reduce the dimension, and add the input and output to form the residual structure. SqueezeNet proposed the Fire module, which replaces the 3×3 convolution kernel with a 1×1 convolution kernel. By adjusting the number of 1×1 convolutions, the number of channels in each layer in the convolution operation can be flexibly controlled, thereby reducing the amount of model calculations. The Shufflenet v1 [15] network, proposed at the same time, is one of the relatively mature lightweight networks. This article uses the improved Shufflenet v2 [16] to initially achieve network lightweighting. In addition to network simplification, numerical calculations have also become a new focus area, with numerical quantification being a typical representative. Quantization is the process of converting a model's weights and activations from floating-point numbers to lower bit-width integers or fixed-point numbers. Quantization can reduce the memory footprint and computational requirements of a model. The literature indicates that model parameters are the primary memory access objects in CNNs; thus, parameter quantization is an effective means of reducing memory access and power consumption. Hardware acceleration involves utilizing specialized hardware accelerators to expedite the inference process of deep learning models, consequently alleviating the computational load on the model.

In summary, for remote sensing object detection, the most effective approach to network lightweighting is to initiate redesign from the feature extraction segment, aiming

to restructure the architecture into a lightweight framework. Consequently, we primarily adopt this methodology, selecting YOLOv5n and Shufflenet v2 as the benchmark network and lightweight architecture, respectively. With additional enhancements in various aspects, we achieved further improvements in model compactness and network performance.

2.3. Attention Mechanism

The attention mechanism, originating from the study of human visual cognition characteristics, represents a significant breakthrough in neural network development. In human visual information processing, due to input information characteristics and human brain processing limitations, it is essential to selectively focus on certain information while ignoring redundant data. For example, in images, focus is on vibrant colors and distinct textures, while in text, attention is on sentence beginnings, endings, and specific keywords. In remote sensing, the attention mechanism adjusts the network's focus on different targets. Moreover, addressing the scarcity of remote sensing datasets, the attention mechanism enriches data, aiding the network in learning more valuable information. Additionally, for multi-source remote sensing images, it integrates diverse information efficiently, enhancing network performance. Therefore, attention mechanisms prioritize which input information to focus on and optimize resource allocation for information processing.

In existing object detection algorithms, the Squeeze-and-Excitation (SE) [33] attention mechanism is widely used in mobile networks. By employing SE modules, the network evaluates relationships between feature channels, determining the importance of each channel through the learning process. It then enhances crucial features for the current task while suppressing less important ones. However, this approach only considers inter-channel information and disregards positional information. Subsequent efforts, like BAM BAM [34] and CBAM [35], aim to incorporate positional information, but convolutional operations capture only local relationships, failing to model crucial long-distance dependencies for visual tasks. To address these limitations and achieve multidimensional information integration, attention mechanisms like Coordinate Attention (CA) and Efficient Channel Attention (ECA) [36] have emerged. Besides focusing on channel and positional information, researchers have explored new approaches, such as evaluating neuron weights and subsequently suppressing or focusing on them based on evaluation results, thus achieving more efficient computations. Representative examples include NAM Attention [37] and SimAM [38]. Furthermore, numerous efficient attention algorithms continue to emerge, including Sequential Attention [39], which emphasizes logical attention, Co-attention [40], which focuses on spatial attention, and the prevalent Transformer based on self-attention.

3. Methods

3.1. The Original YOLOv5n Network

The latest YOLOv5 series includes versions like YOLOv5x, YOLOv5l, YOLOv5m, YOLOv5s, and YOLOv5n. While the model structures of these versions are essentially identical, they are based on YOLOv5l with varying parameter scaling ratios, resulting in differences in model width and depth. Consequently, their network performance varies. Notably, the YOLOv5n network, with the smallest model width and depth, is particularly suitable for devices with stringent real-time performance requirements.

YOLOv5 primarily consists of four parts: Input, Backbone, Neck, and Head. These parts perform functions such as input image preprocessing, feature extraction, feature map fusion, target classification, position regression, and output, as depicted in Figure 1. Each part will be detailed below.

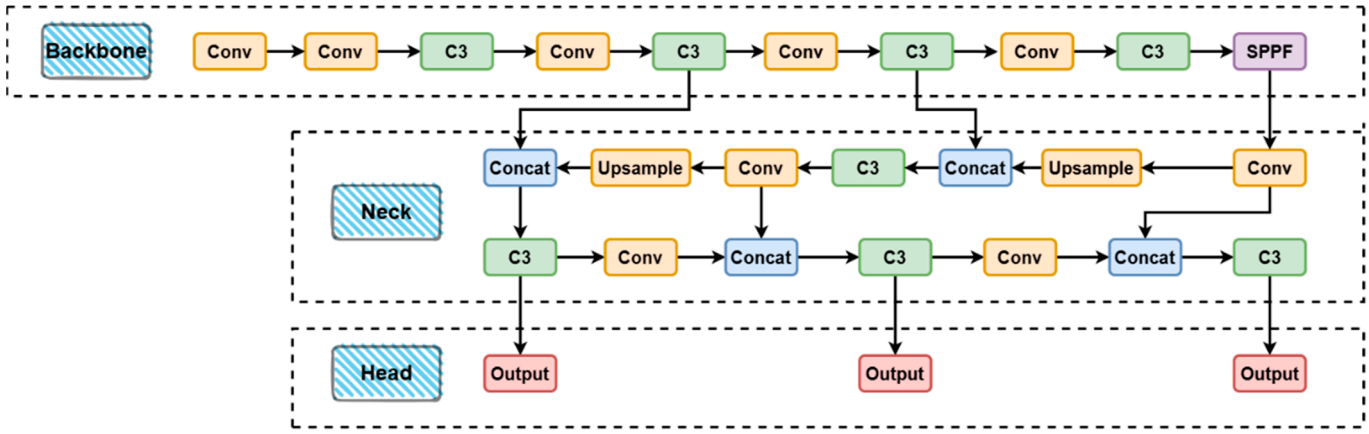

Figure 1. The network structure of YOLOv5.

After inputting the image into the YOLOv5 network, the first step is Mosaic data augmentation. Data augmentation steps typically include random affine transformation, mixing, HSV enhancement, random horizontal flipping, etc., to improve the model's ability to learn and extract target characteristics. The Backbone, responsible for feature extraction, comprises three main modules: Conv, C3, and SPPF. The Conv module consists of convolution, BN, and SiLU activation functions, similar to the previous CBL version, also known as the CBS module in some of the literature. The C3 module architecture resembles the old CSP module, with 3 standard convolutional layers and multiple Bottleneck modules, crucial for learning residual features. The SPPF module, an improvement over the old SPP module, adopts the spatial pyramid concept from the YOLO series. By merging feature maps of different resolutions, it enhances the feature map's expressive ability, particularly beneficial for detecting targets with significant size differences in the image. The Neck module in YOLOv5 combines PAN [41] and FPN [42], positioned centrally within the YOLO network. It serves to bridge the backbone feature extraction segment with the detection head, enhancing the feature map for improved detection accuracy. The Head component primarily receives feature maps generated by the Neck module and processes them to produce category and position information for the detection frame, thus fulfilling the task of target detection.

3.2. Proposed Methods

In this section, we will detail our improvements to the original YOLOv5 network, including Shufflenet, CA, EIoU, and deformable convolution. Figure 2 is the complete structure of our improved network.

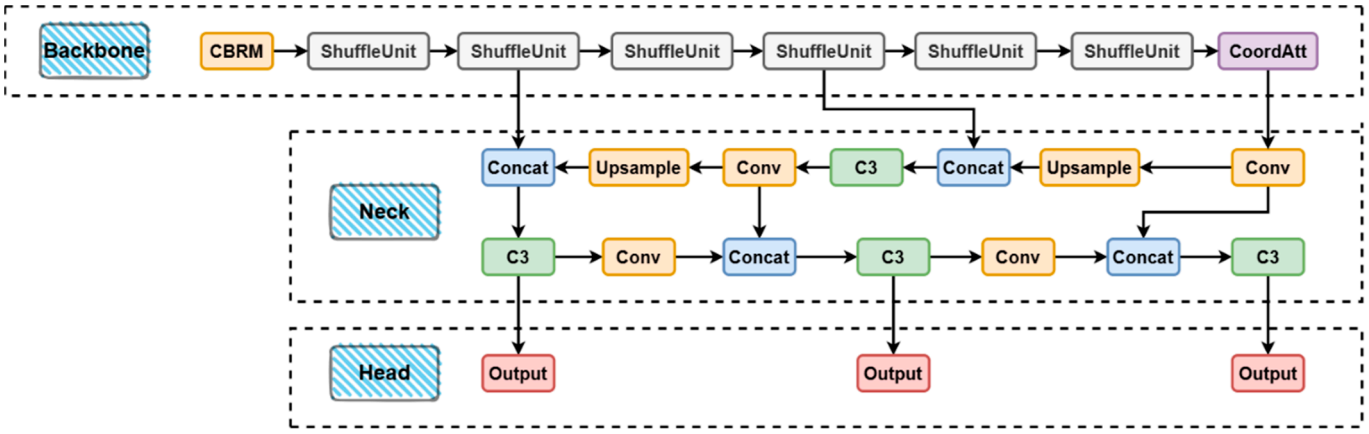

Figure 2. The complete structure of our improved network.

3.2.1. The Redesign of the Backbone

Extensive efforts have been dedicated to optimizing both unit components and the network structure to strike a balance between network volume and detection performance. The Backbone, being pivotal and parameter-intensive in YOLO series algorithms, bears the primary responsibility of extracting features from input images and transmitting them to the neck [41]. Therefore, the design of the Backbone significantly influences the detection accuracy and speed of the entire network. In the original YOLOv5 network, the Backbone typically comprises alternating serial connections of Conv and C3 components. However, our investigation revealed that in the Backbone section, C3 components nearly monopolize all parameters, significantly contributing to the large volume of the original network. Optimizing these modules is essential to achieve network lightweighting. Initially, we defined a new module, CBRM, which sequentially incorporates a convolutional layer followed by batch normalization, ReLU activation, and a max-pooling layer. To achieve a higher cost-effectiveness ratio between network parameters and performance, we drew inspiration from Shufflenet v2, whose structure as below.

We used ShuffleUnit as the constitutive unit for the feature extraction network. ShuffleUnit is a channel shuffle unit designed to enhance the network's learning ability for inter-channel information. This unit operates through two branches, where one branch involves depthwise separable convolution and dimension reduction, while the other branch conducts dimension reduction. When the stride is greater than 1, the outputs of the two branches are concatenated along the channel dimension and then reshuffled through a channel shuffle operation. This reshuffling aids the network in better learning inter-channel feature representations, thereby enhancing overall performance. In summary, ShuffleUnit, through its channel shuffle mechanism, strengthens the network's utilization of diverse channel information, optimizing feature learning and representation. Additionally, we achieve forward propagation by applying a convolutional layer followed by max-pooling to return results from the input tensor. Unlike many similar lightweight networks, for the utmost lightweight design, we abandon the alternating connection of C3 modules and lightweight modules in the original YOLOv5 network and boldly serialize six ShuffleUnits. Extensive research has been conducted on the SPPF in the YOLOv5 network. SPPF, a pyramid-based approach, merges feature maps of different resolutions to enhance the expressive power of the feature map, which proves beneficial when detecting images with significant size differences among objects. However, this approach's drawback lies in its increase in parameters and a significant rise in floating-point operations. Therefore, at the end of the Backbone section, we also abandon the widely used SPPF. While this inevitably leads to some performance loss, we deem it acceptable to achieve our goal of extreme lightweighting.

Overall, the design of ShuffleUnit enables the utilization of more channels at lower computational complexity, thereby enhancing the model's capacity. The feature reuse mode reduces redundancy between feature maps, enhancing accuracy, while its meticulously crafted structure enables ShuffleUnit to strike a balance between speed and accuracy.

3.2.2. Coordinate Attention (CA) Mechanism

The Squeeze-and-Excitation (SE) attention mechanism is widely adopted in mobile networks among existing object detection algorithms. The SE module enables the network to assess the relationship between feature channels, determining the importance of each channel through a learning process. It then enhances significant features relevant to the task while suppressing less critical ones. However, this method solely focuses on inter-channel information and neglects positional information. Subsequent works like BAM and CBAM aimed to incorporate positional information, yet convolution operations could only capture local relationships, failing to model essential long-range dependencies crucial for vision tasks.

CA is an attention mechanism capable of efficiently acquiring inter-channel and positional information simultaneously. It encodes channel relationships and long-range

dependencies using precise positional information. Similar to the SE module, CA consists of two steps: coordinate information embedding and coordinate attention generation [42].

The structure of the CA mechanism is depicted in Figure 3.

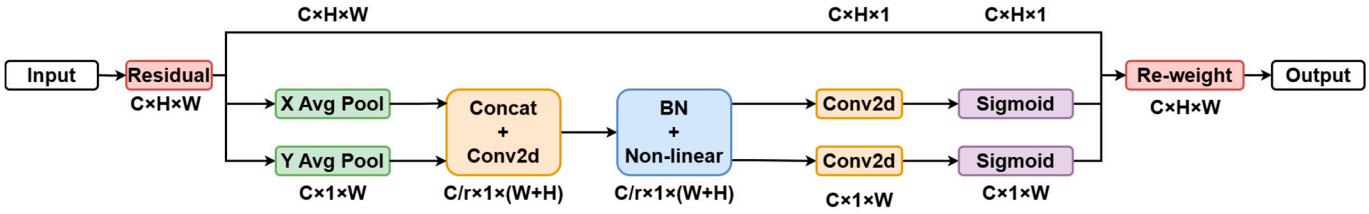

Figure 3. The structure of the CA mechanism.

Global pooling is typically employed in channel attention to encode spatial information globally. However, it compresses global spatial information into one channel descriptor, making it challenging to retain positional information crucial for capturing spatial structure in vision tasks. To address this limitation, CA decomposes the global pooling specified in Equation (1) into two 1D feature encoding operations [43].

$$z_c = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} x_c(i, j) \tag{1}$$

Specifically, given an input X, we utilized two pooling kernels (H, 1) or (1, W) to encode the coordinates of each channel along the horizontal and vertical directions [44]. Aggregating features along two spatial directions separately yielded a pair of feature maps with direction-aware features (see Equations (2) and (3)).

$$z_c^h(h) = \frac{1}{W} \sum_{0 \le i \le W} x_c(h, i) \tag{2}$$

$$z_c^w(w) = \frac{1}{H} \sum_{0 \le i \le H} x_c(j, w) \tag{3}$$

Decomposing channel attention into two parallel vertical and horizontal 1D feature encodings alleviates the loss of position information caused by 2D global pooling, effectively incorporating spatial coordinate information into the generated attention map. Embedding positional information into channel attention offers the advantage of emphasizing large-scale regions without significant computational cost. The algorithm's flexibility and lightweight nature make it easy to port and deploy, significantly enhancing the performance of downstream tasks.

To enable global receptive fields and encode precise position information, CA proposes a coordinate attention generation, guided by three key criteria. First, new transformations should be as simple and economical as possible to be suitable for mobile environments. Second, they should fully utilize captured location information to accurately highlight areas of interest. Finally, they should effectively capture relationships between channels, as demonstrated to be crucial in previous studies. Specifically, given the aggregated feature maps generated by Equations (4) and (5), we first concatenated them and then sent them to the shared 1×1 convolution transformation function F_1, as shown in Equation (4):

$$f = \delta(F_1([z^h, z^w])) \tag{4}$$

This resulted in intermediate feature maps, as shown in Equation (5):

$$f \in R^{C/r \times (H+W)} \tag{5}$$

This intermediate feature map encodes horizontal and vertical spatial information. Here, r is the reduction ratio used to control the block size. We then split f into two independent tensors in the spatial dimension (see Equation (6)).

$$f^h \in R^{C/r \times H} \tag{6}$$

$$f^w \in R^{C/r \times W} \tag{7}$$

In addition, two 1×1 convolution transformations F_h and F_w were used to transform f^h and f^w respectively, and we obtained Equations (8) and (9):

$$g^h = \sigma\left(F_h\left(f^h\right)\right) \tag{8}$$

$$g^w = \sigma(F_w(f^w)) \tag{9}$$

Then, the outputs g^h and g^w were expanded and used as attention weights. Finally, the output y of our coordinate attention block can be written as Equation (10):

$$y_c(i,j) = x_c(i,j) \times g_c^h(i) \times g_c^w(j) \tag{10}$$

3.2.3. EIoU

To address special cases like inclusion and overlap, scholars have introduced Generalized Intersection over Union (GIoU) [45], Distance Intersection over Union (DIoU) and CIoU losses. However, CIoU has limitations due to varying aspect ratios. Our network introduces EIoU (see Equations (11) and (12)):

$$L_{EIoU}\& = L_{IoU} + L_{dis} + L_{asp} = 1 - IoU + \frac{\rho^2\left(b,b^{gt}\right)}{c^2} + \frac{\rho^2(w,w^{gt})}{C_w^2} + \frac{\rho^2\left(h,h^{gt}\right)}{C_h^2} \tag{11}$$

$$L_{Focal-EIoU} = IoU^{\gamma} L_{EIoU} \tag{12}$$

EIOU extends CIoU by incorporating the difference between the prediction and the minimum bounding box size to enhance aspect ratio loss. EIoU takes into account various geometric properties between two bounding boxes, such as width, height, center point distance, and their correlation with the convex hull. EIoU aims to offer a more precise measure of bounding box similarity, particularly in cases of incomplete overlap or significant shape differences. It is employed in object detection tasks to enhance the accuracy of measuring the similarity between object bounding boxes, thereby improving object detection performance. This leads to quicker convergence, enhanced regression accuracy, and the application of focal loss to address sample imbalance and prioritize high-quality anchor boxes during regression.

3.2.4. Deformable Convolutional Network

The Deformable Convolutional Network (DCN) is a CNN variant characterized by the integration of a deformable mechanism into the convolution and Region of Interest (RoI) pooling modules. This mechanism enables the network to learn the sampling position offset to accommodate various task requirements. Unlike traditional fixed convolution, which maintains a static convolution kernel, the deformable convolution dynamically adjusts the position and shape of the kernel at each pixel to match the feature distribution in the input image. This convolution operation is often used in computer vision tasks such as object detection and segmentation because it is better able to capture the characteristics of irregular objects.

Traditional convolution operations employ fixed convolution kernels. In convolution operations, the weights of the convolution kernels remain fixed and cannot adapt to features at various locations and scales. In some cases, the feature distribution may vary greatly in the image, which requires a convolution operation that can be adaptively adjusted

according to the location and shape of the feature. Deformable convolution introduces two key concepts: offset and modulation. The offset vector of each pixel indicates the sampling location in the input image for the convolution kernel. Modulation information dynamically adjusts the shape of the convolution kernel to match various feature shapes. The specific operation of variable convolution is to use offset information to adjust the position of the convolution kernel for each pixel so that it corresponds to a specific position in the input image. Next, the modulation information is used to adjust the shape of the convolution kernel to adapt to the scale and shape of the feature. Finally, a convolution operation is applied at this location to generate the output feature map. Deformable convolution is particularly beneficial in object detection as it can adapt to object shape and position without being constrained by a fixed kernel size and shape. It is also applicable in segmentation tasks to more accurately capture object boundaries and positions. A key feature of Deformable Convolutional Networks is that their inputs and outputs are identical to those of regular convolution and RoI pooling modules. Therefore, they can directly replace traditional modules without altering the existing CNN architecture. Implementing deformable convolution typically involves calculating offset and modulation information. This information can be learned by the network, often through additional convolutional layers that generate offset and modulation information. Additionally, implementing deformable convolution may necessitate custom convolution layers to adjust the convolution kernel based on offset and modulation information.

4. Experimental Design

4.1. Dataset and Evaluation Metrics

To further validate the object detection performance of the proposed network, we selected the publicly available remote sensing image military aircraft target recognition dataset MAR20, which is currently the largest dataset for aircraft target recognition [46]. MAR20 consists of 3842 images, covering 20 aircraft models with 22,341 instances, making it the largest dataset for aircraft target recognition in remote sensing images. The selected aircraft models in this dataset include common types such as SU-35 fighter jets, TU-160 bombers, and TU-22 bombers, resulting in relatively small inter-class differences (see Figure 4).

Figure 4. Examples of MAR20.

Additionally, due to factors such as illumination, atmospheric scattering, and weather conditions, aircrafts of the same category may exhibit different shapes in the dataset, posing challenges for object detection. Furthermore, most images in this dataset are 640 × 640 pixels, with highly diverse target sizes ranging from less than 100 pixels to more than 200 pixels, which further complicates the task of object detection.

To objectively assess the performance of the network model, we evaluated it from two perspectives: model size and detection rate. We defined five parameters for the training process, including model parameter count, model size, detection rate, recall rate, and mAP@0.5. Simultaneously, to assess the object detection speed of the network, we defined four parameters for the detection process: Pre-processing time, inference time, NMS time, and Frames Per Second (FPS).

4.2. Experimental Setup

Table 1 is some information about software, hardware and parameter settings.

Table 1. Information about software, hardware and parameter settings.

GPU	NVIDIA A16 GPU	15 GB RAM
CPU	Intel Xeon Gold 5318Y	
deep learning framework	PyTorch	1.12.0
training iterations		300
batch size		128
initial learning rate		0.01
Final OneCycleLR learning rate		0.1
weight decay		0.01
momentum		0.937

4.3. Performance Comparison of Baseline Models and Enhanced Models

4.3.1. Training Results

To comprehensively compare the performance of this lightweight network, we selected GhostNet [13], PP-LCNet [14], MobileNetV3, and MobileNetV3s as benchmark models. The experiments were conducted by replacing the backbone network of YOLOv5n with each of these models. The experimental results are presented in Table 2.

Table 2. Experimental results of replacing the backbone network of YOLOv5n with each of these models.

	Parameters	**Weights**	**P**	**R**	**mAP@0.5**
YOLOv5n	1,790,977	3.78 M	0.968	0.950	0.979
Ghost-Net	2,413,849	19.3 M	0.924	0.925	0.952
PP-LCNet	248,597	0.9 M	0.768	0.795	0.844
Mobile-NetV3	2,507,231	19.9 M	0.936	0.925	0.967
MobileNetV3s	816,591	2.0 M	0.916	0.869	0.936
Proposed	408,503	1.13 M	0.912	0.913	0.952

The results above demonstrate a notable reduction in parameter count, model size, and floating-point computations compared to the original YOLOv5n network structure. Compared to PP-LCNet and MobileNetV3s, which have similar model sizes, our model exhibits advantages in both performance and model size. Despite being nearly 20 times larger in model size than MobileNetV3 and Ghost-Net, our model achieves a size reduction of almost 20 times while sacrificing only 1.5% of performance compared to MobileNetV3, and even performing comparably to Ghost-Net.

It is important to note that, in striving for the optimal balance between detection performance and resource consumption, we made the reluctant decision to sacrifice a

2.7-percentage-point improvement in detection efficiency. The detection rate is evidently a crucial evaluation metric in remote sensing object detection networks. Historically, we have devoted considerable effort solely to improving the detection rate. We have also made related attempts, such as using more sophisticated network structures to improve the detection rate, with notable examples being the addition of the pyramid structure ASPP (Atrous Spatial Pyramid Pooling). However, after extensive experimentation and analysis, we discovered that, for remote sensing object detection networks, many of our efforts are marginal once the detection rate reaches a certain high level (e.g., around 0.95). In practical terms, the differences are not significant, but the increase in memory and computational consumption due to more complex network structures is often rapid. Our network resulted in a reduction in the number of parameters to approximately one-fifth of the original YOLOv5n network and a decrease in model file size to approximately one-third. This is highly advantageous for deployment on lightweight mobile platforms. Therefore, we believe that the 2.7% loss in detection rate is worthwhile. In the following experiments, we will further study how to narrow the gap in detection rate with the original network as much as possible while maintaining the current memory and computing power consumption.

Consequently, our network attains an excellent balance between model size and performance.

4.3.2. Detecting Results

To further evaluate the network's detection speed, we tested it using the MAR20 dataset on an embedded NVIDIA Jetson Xavier NX device, comparing it with the aforementioned networks. The platform configuration can be found in Table 3.

Table 3. Information of embedded device.

CPU	Carmel ARM®v8.2	8 GB
GPU	NVIDIA Volta™ GPU	8 GB
Computing Power		21TOPS

For experimental results, refer to Table 4.

Table 4. Detecting results of replacing the backbone network of YOLOv5n with each of these models.

	Pre-Process	**Inference**	**NMS**	**FPS**
YOLOv5n	1.6 ms	41.0 ms	4.1 ms	21.4
GhostNet	1.5 ms	51.8 ms	3.5 ms	17.6
PP-LCNet	2.1 ms	30.0 ms	4.0 ms	27.7
MobileNetV3	1.5 ms	64.5 ms	4.1 ms	14.3
MobileNetV3s	1.7 ms	40.1 ms	3.6 ms	22.0
Proposed	2.5 ms	37.6 ms	4.1 ms	22.6

Figure 5 provides examples of testing the same image in MAR20 using both our network and YOLOv5n. Our network can identify the position and categories of the aircraft target even when it is occluded or overlapped, relying on its local characteristics. To verify our detection results, we also display the detection results of YOLOv5n on the same images in Figure 5. The comparison results show that the position and category information of the aircraft target identified by the local features at the edge of the image are all correct.

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

Figure 5. Examples of using our network and YOLOv5n to test the same image in MAR20. (**a**): YOLOv5n; (**b**): our network.

Figure 6 demonstrates that our network can effectively separate dense small aircraft targets and accurately perform position and classification regression.

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

Figure 6. (**a**–**d**) illustrate examples of dense small aircraft objects in images from several distinct scene backgrounds, tested using our network on the MAR20 dataset.

In certain scenarios, YOLOv5n exhibits excessive sensitivity to local target characteristics, resulting in false detections where areas with similar features at the image edge are misclassified as aircraft targets. Our network mitigates this issue, as shown in Example Figure 7.

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

Figure 7. Examples of using our network and YOLOv5n to test the same image in MAR20. (**a**): YOLOv5n, and which has misjudged aircraft targets; (**b**): our network.

For certain aircraft categories with small target sizes and less distinct features in certain remote sensing images, such as A20, YOLOv5n is prone to missed detections, whereas our network has addressed this issue, as depicted in Figure 8.

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

Figure 8. Examples of using our network and YOLOv5n to test the same image which have less obvious features in MAR20. (**a**): YOLOv5n, and which has missed aircraft targets; (**b**): our network.

Although there are discrepancies between our network and the original YOLOv5n in metrics such as mAP@0.5, the image test results demonstrate that our network accurately captures the position of each aircraft target in object detection. For the fine-grained classification of aircraft with similar characteristics, such as A11, A18 and A20, the probability of being misjudged will increase, which will be among the main focal points for our future improvement endeavors.

4.4. Results of Ablation Experiments

To further validate the effectiveness of our innovations, we also conducted ablation experiments on the proposed network. Table 5 provides the specific ablation experiment comparison network settings.

Table 5. Detailed settings of ablation experiment.

	YOLOv5n	CA	ShuffleUnit	EIoU	Deformable Conv
4.4-a	✔				
4.4-b	✔	✔			
4.4-c	✔		✔		
4.4-d	✔	✔	✔		
4.4-e	✔	✔	✔	✔	
Proposed	✔	✔	✔	✔	✔

Also, using Mar20 as the experimental dataset, we have listed the experimental results in Table 6.

Table 6. Results of ablation experiment.

	Parameters	Weights	P	R	mAP@0.5
4.4-a	1,790,977	3.78 M	0.968	0.950	0.979
4.4-b	1,799,335	3.80 M	0.967	0.957	0.982
4.4-c	229,889	0.80 M	0.689	0.732	0.784
4.4-d	278,625	0.88 M	0.896	0.866	0.927
4.4-e	279,621	0.90 M	0.909	0.881	0.936
Proposed	408,503	1.13 M	0.912	0.913	0.952

The experimental results in the table above clearly indicate that each innovation we introduced has had a certain impact compared to the original network. However, for the

fine-grained classification of aircrafts with similar features, such as A11, A18, and A20, the probability of misclassification increases. This will be a primary focus of our future improvement efforts.

4.5. Results of Generalization Experiments

We conducted a series of experiments to verify the generalization of the proposed method and to evaluate its performance in detecting various remote sensing targets. Besides aircraft targets, we detected three common remote sensing targets, oil tanks, cars, and trucks, for remote sensing image target detection [47,48]. The experimental results demonstrate the effectiveness of our method in detecting these targets, achieving a high detection rate. Specifically, the detection rate for oil tank targets was 93.8%, and for car and truck targets, it was 99.5% and 98.9%, respectively. Therefore, our network can train and detect small-sized objects well in remote sensing images such as vehicles and aircraft. These results further confirm the effectiveness and generalizability of our proposed method, indicating its potential application in various remote sensing target detection tasks. Therefore, we are confident in the scalability and applicability of the proposed method and anticipate further exploration of its performance and application across a broader range of remote sensing scenarios in future research.

The experimental results can be found in Figure 9.

Figure 9. Examples of generalization experiments on oil tanks, cars and trucks.

At the same time, we also list some detailed experimental data in Table 7. It can be seen that whether it is Recall or Precision, our experimental results are very satisfactory.

Table 7. Detecting results of replacing the backbone network of YOLOv5n with each of these models.

Class	P	R	mAP@0.5
Oil tanks	0.667	0.908	0.938
Cars	0.985	1	0.995
Trucks	1	0.815	0.989

5. Conclusions

This article introduces an ultra-lightweight remote sensing aircraft object detection network based on YOLOv5n. To balance detection accuracy and speed, we draw inspiration from Shufflenet v2. Utilizing grouped convolution and channel shuffling methods, we constructed ShuffleUnit as the component unit of the feature extraction network. ShuffleUnit defines a deep convolution operation and implements group convolution through the connection of two branches and the rearrangement of channels, thereby improving the nonlinear expression ability of the network and reducing the amount of calculation. Additionally, we implemented forward propagation to include channel splitting, branch calculation, and channel rearrangement. In contrast to many similar lightweight networks, we opted for an extreme lightweight approach by abandoning the alternate connection of C3 modules and lightweight modules in the original YOLOv5 network, and instead, we

boldly connected six shuffleblocks in series. Furthermore, to further enhance the network's performance, we embedded the CA attention mechanism at the end of the backbone. CA is an attention mechanism that can efficiently obtain inter-channel information and position information at the same time. The CA module encodes channel relationships and long-range dependencies with precise location information [21], thereby helping the network locate objects of interest more accurately. Additionally, we replaced the CIoU in the original network with the more advanced EIoU for the loss function, enhancing the aspect ratio loss by considering the difference between the prediction and the minimum bounding box size. Finally, to better handle deformation for irregular areas and targets, we optimized some traditional convolutions into deformable convolutions, resulting in a reduction in the number of parameters.

Experimental results on the public dataset MAR20 demonstrate that our lightweight network, compared to the original YOLOv5n network, has approximately one-fifth of its parameter count, achieving a 6.5% increase in Frames Per Second (FPS) with only a slight 2.7% decrease in mAP@0.5. Moreover, it has a compact model volume of only 1.13 MB. After deployment on the embedded device, the network achieved an FPS of 22.8 even with relatively limited computing power, enabling real-time object detection. Finally, based on YOLOv5n, experiments were conducted on models with lightweight networks such as Shufflenet v2, GhostNet, PP-LCNet, MobileNetV3, and MobileNetV3s as the backbone network, comparing detection accuracy, model size, and detection speed. The results achieved the optimal balance, confirming the feasibility and superiority of the network. Furthermore, a series of ablation experiments were conducted on the proposed network to further validate the effectiveness of our innovations. The results show that each innovation point added produces a certain degree of effect compared to the original network. According to the experimental results, our network accurately captured the position of each aircraft target in the position regression problem of object detection. For the fine-grained classification of aircraft with similar characteristics, such as A11, A18, and A20, the probability of misjudgment increases, which will be a primary focus of our future improvement efforts.

Author Contributions: J.W. and Z.B. conceived and designed the experiments; J.W. and X.Z. performed the experiments; Z.B. and X.Z. analyzed the data; J.W. wrote the paper; Y.Q. supervised this work. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by national defense foundation, grant number E21D41B1.

Data Availability Statement: The project presented in this study are available through: https://github.com/wanxxiaxe/lightweightOB_2nd (accessed on 8 January 2024).

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

References

1. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 580–587.
2. Girshick, R. Fast r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
3. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *arXiv* **2015**, arXiv:1506.01497. [CrossRef] [PubMed]
4. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
5. Redmon, J.; Farhadi, A. YOLO9000: Better, faster, stronger. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 7263–7271.
6. Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
7. Bochkovskiy, A.; Wang, C.-Y.; Liao, H.-Y.M. Yolov4: Optimal speed and accuracy of object detection. *arXiv* **2020**, arXiv:2004.10934.
8. Jocher, G. YOLOv5 by Ultralytics. Available online: https://github.com/ultralytics/yolov5 (accessed on 1 August 2023).
9. Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; Zagoruyko, S. End-to-end object detection with transformers. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; pp. 213–229.

10. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv* **2017**, arXiv:1704.04861.
11. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.-C. Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4510–4520.
12. Howard, A.; Sandler, M.; Chu, G.; Chen, L.-C.; Chen, B.; Tan, M.; Wang, W.; Zhu, Y.; Pang, R.; Vasudevan, V. Searching for mobilenetv3. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 1314–1324.
13. 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, 13–19 June 2020; pp. 1580–1589.
14. Cui, C.; Gao, T.; Wei, S.; Du, Y.; Guo, R.; Dong, S.; Lu, B.; Zhou, Y.; Lv, X.; Liu, Q. PP-LCNet: A lightweight CPU convolutional neural network. *arXiv* **2021**, arXiv:2109.15099.
15. Zhang, X.; Zhou, X.; Lin, M.; Sun, J. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 6848–6856.
16. Ma, N.; Zhang, X.; Zheng, H.-T.; Sun, J. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 116–131.
17. Jiang, Y.; Tang, Y.; Ying, C.J.E. Finding a Needle in a Haystack: Faint and Small Space Object Detection in 16-Bit Astronomical Images Using a Deep Learning-Based Approach. *Electronics* **2023**, *12*, 4820. [CrossRef]
18. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *ACM* **2012**, *60*, 84–90. [CrossRef]
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 Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016; Part I 14, pp. 21–37.
20. Deng, Z.; Sun, H.; Zhou, S.; Zhao, J.; Lei, L.; Zou, H. Multi-scale object detection in remote sensing imagery with convolutional neural networks. *ISPRS J. Photogramm. Remote Sens.* **2018**, *145*, 3–22. [CrossRef]
21. Cheng, S.; Cheng, H.; Yang, R.; Zhou, J.; Li, Z.; Shi, B.; Lee, M.; Ma, Q.J.P. A High Performance Wheat Disease Detection Based on Position Information. *Plants* **2023**, *12*, 1191. [CrossRef]
22. Luo, S.; Yu, J.; Xi, Y.; Liao, X.J.I.A. Aircraft target detection in remote sensing images based on improved YOLOv5. *IEEE Access* **2022**, *10*, 5184–5192. [CrossRef]
23. Liu, Z.; Gao, Y.; Du, Q.; Chen, M.; Lv, W.J.I.A. YOLO-extract: Improved YOLOv5 for aircraft object detection in remote sensing images. *IEEE Access* **2023**, *11*, 1742–1751. [CrossRef]
24. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *arXiv* **2017**, arXiv:1706.03762.
25. Liu, L.; Pan, Z.; Lei, B. Learning a rotation invariant detector with rotatable bounding box. *arXiv* **2017**, arXiv:1711.09405.
26. Yang, X.; Yan, J.; Feng, Z.; He, T. R3det: Refined single-stage detector with feature refinement for rotating object. In Proceedings of the AAAI Conference on Artificial Intelligence, Virtually, 2–9 February 2021; pp. 3163–3171.
27. 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.
28. Zhou, L.; Wei, H.; Li, H.; Zhao, W.; Zhang, Y.; Zhang, Y. Arbitrary-oriented object detection in remote sensing images based on polar coordinates. *IEEE Access* **2020**, *8*, 223373–223384. [CrossRef]
29. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
30. Iandola, F.N.; Han, S.; Moskewicz, M.W.; Ashraf, K.; Dally, W.J.; Keutzer, K. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and< 0.5 MB model size. *arXiv* **2016**, arXiv:1602.07360.
31. Gholami, A.; Kwon, K.; Wu, B.; Tai, Z.; Yue, X.; Jin, P.; Zhao, S.; Keutzer, K. Squeezenext: Hardware-aware neural network design. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Salt Lake City, UT, USA, 18–23 June 2018; pp. 1638–1647.
32. Sifre, L.; Mallat, S. Rigid-motion scattering for texture classification. *arXiv* **2014**, arXiv:1403.1687.
33. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7132–7141.
34. Park, J.; Woo, S.; Lee, J.-Y.; Kweon, I.S. Bam: Bottleneck attention module. *arXiv* **2018**, arXiv:1807.06514.
35. Woo, S.; Park, J.; Lee, J.-Y.; Kweon, I.S. Cbam: Convolutional block attention module. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 3–19.
36. Wang, Q.; Wu, B.; Zhu, P.; Li, P.; Zuo, W.; Hu, Q. ECA-Net: Efficient channel attention for deep convolutional neural networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 11534–11542.
37. Liu, Y.; Shao, Z.; Teng, Y.; Hoffmann, N. NAM: Normalization-based attention module. *arXiv* **2021**, arXiv:2111.12419.
38. Yang, L.; Zhang, R.-Y.; Li, L.; Xie, X. Simam: A simple, parameter-free attention module for convolutional neural networks. In Proceedings of the International Conference on Machine Learning, Virtual, 18–24 July 2021; pp. 11863–11874.

39. Chen, Q.; Wang, W. Sequential attention-based network for noetic end-to-end response selection. *arXiv* **2019**, arXiv:1901.02609.
40. Feng, G.; Hu, Z.; Zhang, L.; Lu, H. Encoder fusion network with co-attention embedding for referring image segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 15506–15515.
41. Han, S.; Jiang, X.; Wu, Z. An improved YOLOv5 algorithm for wood defect detection based on attention. *IEEE Access* **2023**, *11*, 71800–71810. [CrossRef]
42. Qiu, S.; Li, Y.; Zhao, H.; Li, X.; Yuan, X. Foxtail Millet Ear Detection Method Based on Attention Mechanism and Improved YOLOv5. *Sensors* **2022**, *22*, 8206. [CrossRef] [PubMed]
43. Shi, Y.; Gao, Z.; Li, S. Real-Time Detection Algorithm of Marine Organisms Based on Improved YOLOv4-Tiny. *IEEE Access* **2022**, *10*, 131361–131373. [CrossRef]
44. Wang, M.; Zhu, Y.; Liu, Y.; Deng, H.J.S.; Networks, C. X-Ray Small Target Security Inspection Based on TB-YOLOv5. *Secur. Commun. Netw.* **2022**, *2022*, 2050793. [CrossRef]
45. Rezatofighi, H.; Tsoi, N.; Gwak, J.; Sadeghian, A.; Reid, I.; Savarese, S. Generalized intersection over union: A metric and a loss for bounding box regression. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 658–666.
46. Yu, W.; Cheng, G.; Wang, M.; Yao, Y.; Xie, X.; Yao, X.; Han, J. MAR20: A Benchmark for Military Aircraft Recognition in Remote Sensing Images. *Natl. Remote Sens. Bull.* **2022**, *27*, 2688–2696. [CrossRef]
47. Li, K.; Wan, G.; Cheng, G.; Meng, L.; Han, J. Object detection in optical remote sensing images: A survey and a new benchmark. *ISPRS J. Photogramm. Remote Sens.* **2020**, *159*, 296–307. [CrossRef]
48. Akson, N. teknofest2 Dataset. Roboflow Universe. Available online: https://universe.roboflow.com/neslihan-akson-rvo7y/teknofest2-ftdcf (accessed on 31 January 2024).