applied sciences

MDPI

Article

Improved YOLOv7 Algorithm for Small Object Detection in Unmanned Aerial Vehicle Image Scenarios

Xinmin Li [1,2], Yingkun Wei [3], Jiahui Li [3], Wenwen Duan [3], Xiaoqiang Zhang [3,*] and Yi Huang [4]

[1] College of Computer Science, Chengdu University, Chengdu 610100, China; lxm_edu@126.com
[2] Guangdong Provincial Key Laboratory of Future Networks of Intelligence, The Chinese University of Hong Kong, Shenzhen 518172, China
[3] School of Information Engineering, Southwest University of Science and Technology, Mianyang 621000, China
[4] Department of Information and Communication Engineering, Tongji University, Shanghai 201804, China; huangyi718b@tongji.edu.cn
* Correspondence: xqzhang@swust.edu.cn

Abstract: Object detection in unmanned aerial vehicle (UAV) images has become a popular research topic in recent years. However, UAV images are captured from high altitudes with a large proportion of small objects and dense object regions, posing a significant challenge to small object detection. To solve this issue, we propose an efficient YOLOv7-UAV algorithm in which a low-level prediction head (P2) is added to detect small objects from the shallow feature map, and a deep-level prediction head (P5) is removed to reduce the effect of excessive down-sampling. Furthermore, we modify the bidirectional feature pyramid network (BiFPN) structure with a weighted cross-level connection to enhance the fusion effectiveness of multi-scale feature maps in UAV images. To mitigate the mismatch between the prediction box and ground-truth box, the SCYLLA-IoU (SIoU) function is employed in the regression loss to accelerate the training convergence process. Moreover, the proposed YOLOv7-UAV algorithm has been quantified and compiled in the Vitis-AI development environment and validated in terms of power consumption and hardware resources on the FPGA platform. The experiments show that the resource consumption of YOLOv7-UAV is reduced by 28%, the mAP is improved by 3.9% compared to YOLOv7, and the FPGA implementation improves the energy efficiency by 12 times compared to the GPU.

Keywords: UAV images; small object detection; YOLOv7; FPGA; Vitis-AI

Citation: Li, X.; Wei, Y.; Li, J.; Duan, W.; Zhang, X.; Huang, Y. Improved YOLOv7 Algorithm for Small Object Detection in Unmanned Aerial Vehicle Image Scenarios. *Appl. Sci.* **2024**, *14*, 1664. https://doi.org/10.3390/app14041664

Academic Editor: Gianluigi Ferrari

Received: 4 January 2024
Revised: 7 February 2024
Accepted: 16 February 2024
Published: 19 February 2024

1. Introduction

Object detection is a significant research topic in image processing and computer vision and enables the development of advanced computer vision applications [1,2]. In recent years, owing to profound advancements in deep learning research, intelligent object detection systems have found extensive applications in diverse domains, including automatic driving, obstacle recognition, video surveillance, medical imaging, and virtual reality [3–6]. In particular, with the advantages of flexibility and maneuverability, unmanned aerial vehicles (UAVs) are regarded as a potential technology for both commercial and military applications, including disaster surveillance, traffic patrol, aerial base stations, and the navigation of military battlefields [7–9]. As mobile platforms at high altitudes, UAVs can acquire photos more quickly, flexibly, and economically by traversing the area of interest, offering strong support for subsequent information processing. Investigating precious object detection in UAV image scenarios is an ongoing research topic due to the high altitudes of UAVs.

UAV images are captured by UAVs at high altitudes, resulting in image characteristics markedly distinct from those of natural images [10]. First, the proportion of small objects in UAV images is very high compared to natural image datasets due to their high altitudes. Second, the size and shape of ground objects vary depending on the height and angle

of the UAV observation. Third, the objects tend to gather in certain areas and are often obscured from each other. Although the existing object detection algorithms enable the effective detection of common objects in natural scenes, it is difficult to detect small objects in UAV images since they are present in large proportions and occupy fewer pixels than regular objects [11,12]. Meanwhile, UAVs have limited energy and hardware resources to carry out the algorithm [13,14]. First, the general power consumption of the standard UAV is relatively low. Because of the need for a lightweight structure and prolonged endurance, it is difficult to carry high-power-consumption platforms for implementing deep learning algorithms on UAVs. Second, image processing involves numerous pixel and matrix operations, and the hardware resources of the deployed platform are related to the quality and speed of processing. Memory is tasked with both storing image data and facilitating temporary transfers. The data bus is responsible for transmitting pixel data, and the bandwidth affects the transmission of image data at the same time [15]. These existing problems make object detection in UAV images a significant challenge in practical applications.

To address these issues, we propose the efficient YOLOv7-UAV algorithm for small object detection in UAV image scenarios. The algorithm has been deployed on a low-power FPGA platform. Compared to our previous research [16], this paper aims to improve the effectiveness of multi-scale feature fusion based on the proposed BiFPN-like structure. The main contributions of this work are summarized as follows:

- We propose the YOLOv7-UAV algorithm to detect small objects in UAV image scenarios, in which the low-level prediction head P2 is added to improve the feature extraction performance for small objects and the deep-level prediction head P5 is removed to reduce the impact of deep features.
- The BiFPN-like structure is adopted to fuse the semantic and geometric feature information more efficiently by a weighted cross-level connection. Taking the direction mismatch between the prediction box and the ground-truth box into account, the SIoU function is used for the regression loss to improve the performance in fitting the prediction box to the ground-truth box.
- The proposed YOLOv7-UAV algorithm was quantified and compiled in Vitis-AI for deployment on FPGA to verify the power and hardware resource consumption. On the VisDrone-2019 dataset, our proposed YOLOv7-UAV achieves 45.3% (mAP), an improvement of 3.9%, and reduces the resource consumption by 28% compared to the traditional YOLOv7 algorithm. The power consumption is decreased by more than 200 W, improving energy efficiency by 12 times compared to GPU schemes.

2. Related Work

2.1. YOLOv7 Algorithm

Accuracy and speed serve as the basic performance metrics in deep learning-based object detection algorithms [17]. Fast-RCNN [18] and Mask-RCNN [19] are typical two-stage algorithms that can improve the accuracy. In the first stage, the suggested areas are obtained from the input image. In the second stage, these regions are sent to the classifier to determine the category and confidence level of objects. However, the two-stage network requires significant computational resources, making it unsuitable for deployment on UAVs that require real-time detection [20]. Redmon et al. [21] firstly proposed the YOLO algorithm with a one-stage model, which used a single neural network to output the predicted bounding box and the corresponding confidence levels of objects. The YOLO algorithms are characterized by a lightweight network architecture that balances speed and accuracy, making them widely used for real-time object detection on edge devices [22,23].

Among the studies of YOLO algorithms, Wang et al. [24] proposed the YOLOv7 algorithm and implemented a range of strategies to enhance its performance based on an extended efficient long-range attention network (E-ELAN), max pooling convolution (MPConv), and planned re-parameterized convolution (RepConv) strategies. In particular, the E-ELAN structure adopts group convolution to expand the channel and cardinality

of computational blocks to enhance the learning ability. With the above effective network architectures, YOLOv7 outperformed the existing object detection algorithm in speed and accuracy in a range from 5 FPS to 160 FPS [24]. Due to its significant advantages, YOLOv7 is suitable for deployment on edge devices such as UAVs. Therefore, we aim to improve the performance of small object detection in UAV image scenarios based on the traditional YOLOv7.

2.2. Small Object Detection

For small object detection scenarios, Zhu et al. [25] added a prediction head and replaced all prediction heads with the transformer prediction head structure to enhance the performance in detecting small objects. Li et al. [26] used a cross-layer attention network to obtain better features of small objects. Deng et al. [27] acquired high-resolution features for small object detection by deconvolution and sub-pixel convolution. Chen et al. [28] used a shuffle attention structure and fused different feature maps to enhance the ability to perform multi-scale detection. The above methods were developed by adding network structures to enhance the detection of small objects. However, this inevitably results in an increase in the training time and model parameters. Consequently, they are not suitable for UAVs due to their limited energy and hardware resources.

Enhancing the performance of small object detection is achievable through the optimization of the feature enhancement and detection framework [11,29–31]. Liu et al. [11] proposed the single-shot multi-box detector (SSD), which uses multiple feature maps to predict objects at different scales. The shallow feature map is responsible for predicting small objects due to its high-resolution in the SSD. This is based on the consideration that shallow features with rich geometric information are very useful for the detection of small objects. Redmon et al. [29] employed multi-scale prediction by adding branches, and the high-resolution features are responsible for small objects in YOLOv3. Yang et al. [30] added a prediction head to better retain small object feature information. By analyzing the relationship between the down-sampling of the input images and the features of small objects, Xue et al. [31] removed the network structure of the deep-level detection head to reduce the effect of the features of small objects. As shown in Figure 1, the UAV image is down-sampled several times; Figure 1a has the lowest resolution due to multiple down-sampling processes, and Figure 1c has the highest resolution with less down-sampling. This indicates that the low-resolution feature map corresponds to large anchor boxes, which are more difficult to match to the ground-truth boxes of small objects. Meanwhile, the low-resolution feature map corresponds to the largest receptive field, causing some specific information about small objects to be lost.

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

Figure 1. Visual prediction graph. (**a**–**c**) The input images are shown to be down-sampled to 5 × 5, 10 × 10, and 15 × 15, respectively.

3. Methodology

3.1. Network Structure

The structure of the proposed YOLOv7-UAV network is shown in Figures 2 and 3. The proposed network mainly consists of three parts: the input, backbone, and head. Different

from the traditional YOlOv7, the prediction results for the YOLOv7-UAV network originate from the three prediction heads: P2, P3, and P4.

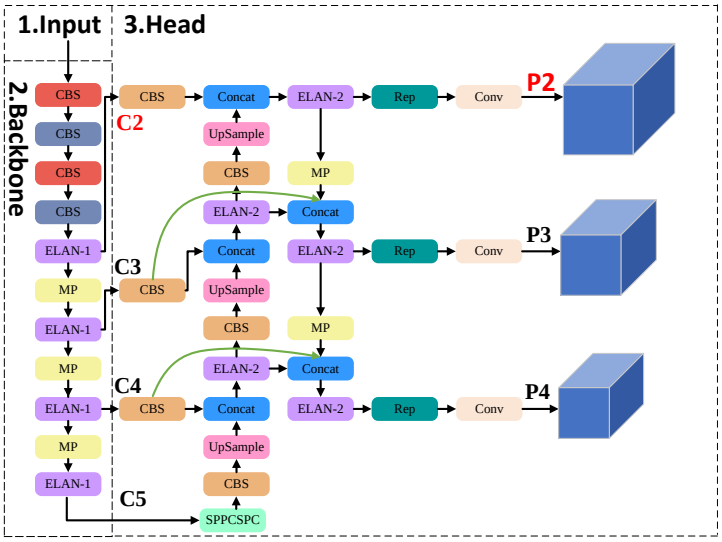

Figure 2. Overall architecture of YOLOv7-UAV.

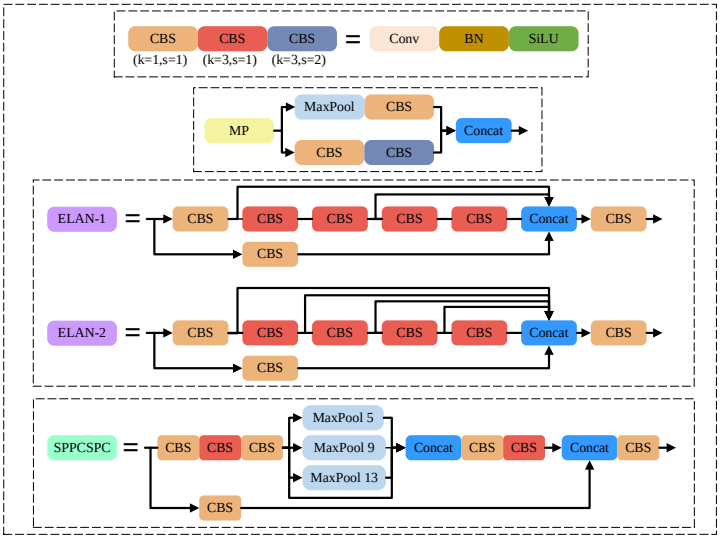

Figure 3. The modules of YOLOv7-UAV.

In order to extract features from the input images in the backbone structure, we first adopt the multiple convolution blocks, E-ELAN, and MP structures of the conventional YOLOv7 network. As shown in Figure 2, there are four different scale sizes of the feature maps extracted by the backbone network: C2, C3, C4, and C5. C2 represents the feature map generated in the second stage of the backbone network, where the input image resolution is halved two times. Similarly, C5 represents the feature map generated in the fifth stage of the backbone network, where the resolution is halved five times. C2 is a newly introduced shallow feature map with high resolution that facilitates the detection of small objects with detailed information in UAV images. Second, comparable network units from previous prediction heads are similarly used by the new P2 prediction head and its associated network. Third, the traditional P5 prediction head and its associated network units are eliminated. Finally, a BiFPN-like feature fusion network is added, and the edge nodes from C2 are retained.

As shown in Figure 3, the YOLOv7-UAV network mainly has CBS, ELAN, MP, SPPCSPC, and RepConv modules. First, the CBS module consists of convolution (Conv), batch

normalization (BN), and a sigmoid linear unit (SiLU). Different colors represent different kernel sizes and step sizes of convolution: e.g., (k = 1, s = 2) means that the kernel size k = 1 and the step size s = 2. Second, the ELAN module is composed of several CBSs and uses expand, shuffle, and merge cardinality to achieve the ability to improve the learning ability of the network without destroying the original gradient path [24]. Third, the MP module mainly consists of max pooling and CBS, which improves the ability of the model to extract and merge features. Fourth, the SPPCSPC module enriches the feature information by obtaining different receptive fields through multiple max pooling operations and merging them with the previous feature maps through the concat operation. Finally, the RepConv module uses a special residual structure to support training. In the prediction process of the network, 3 × 3 convolution is used for the output to reduce the complexity of the network and make it easier to deploy. Based on these efficient network structures, YOLOv7-UAV can obtain rich feature information to detect small objects in UAV images.

3.2. Extra Prediction Head (P2)

In the complex background of UAV images, objects on the ground are difficult to detect accurately due to their small size or the dense object environment. Therefore, we add an extra detection head based on the traditional YOLOv7 network to better detect small objects in UAV images. The prediction result of the new prediction head is generated from the shallow feature maps, providing richer feature information on small objects than the deep-level feature maps.

As shown in Figure 4, the new shallow feature map has more feature information. The C2, C3, C4, and C5 feature maps are extracted by the backbone network with 4, 8, 16, and 32 times down-sampling, respectively. The deepest feature map obtained by multiple convolutions corresponds to the P5 prediction head. Thus, P5 has the largest anchor box size and receptive field, which make it difficult to fit the prediction box to the ground-truth box and extract the feature information of small objects. In addition, multiple operations of down-sampling of the input images will affect the features of small objects in UAV images with many small objects. The prediction head P2 from the shallow feature map corresponds to the small anchor box size and receptive field, which can be effective in extracting the feature information of small objects. In order to enhance small object detection, the P2 prediction head is used by the YOLOv7-UAV network in place of the P5 prediction head. Meanwhile, we cut off the part of the network units containing the P5 detection head. In addition, the extra P2 prediction head increases the parameters in the network, but the removed deep-level P5 prediction head can reduce the number of parameters even more. If we adopt the lower-level prediction head P1 with a resolution of 320 × 320, the shallower feature map C1 will be used as the basic feature map to fuse other feature maps. C1 undergoes only one stage of feature extraction, resulting in inadequate feature extraction and an inability to accurately predict objects. Therefore, we chose the extra prediction head P2 instead of the lower-level prediction head P1.

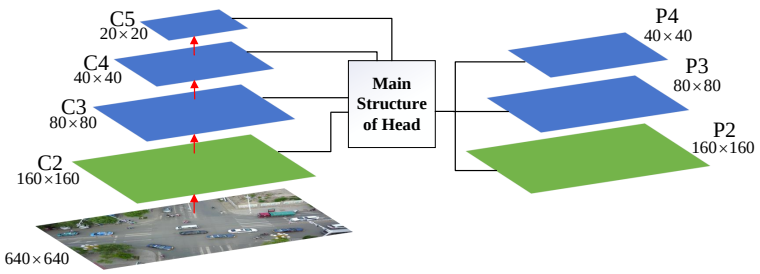

Figure 4. Feature extraction model of YOLOv7-UAV.

An object detector that is based on anchors is sensitive to the size of the anchor box. Therefore, the K-means++ algorithm is used to adjust the size of the anchor box for small objects [32]. Table 1 shows the correspondence between the three prediction heads and the

anchor box sizes for the three scales when the image size is 640 × 640 on the VisDrone2019 dataset [33].

Table 1. Anchor box size setting in YOLOv7-UAV.

Prediction Head	Feature Graph	Anchor Size
P2	160 × 160	(3,4) (3,8) (7,6)
P3	80 × 80	(6,13) (14,8) (12,17)
P4	40 × 40	(27,14) (21,29) (48,38)

3.3. Feature Fusion Framework

The feature fusion structure adopted by the traditional YOLOv7 integrates principles from both the feature pyramid network (FPN) [34] and the path aggregation network (PANnet) [35]. Top-down feature fusion transfers rich semantic information from the high-level layers to the low-level layers by up-sampling. Meanwhile, bottom-up feature fusion transfers rich geometric information from the low level to the high level by down-sampling. This bidirectional feature fusion structure has demonstrated promising results in the feature fusion of conventional ground images. However, UAV images pose a greater challenge due to the abundance of small objects and the distribution of multi-scale objects.

The BiFPN framework is anticipated to address the above challenges [36]. Compared with the traditional PANet feature fusion structure of the YOLOv7 algorithm, BiFPN not only includes up-sampling and down-sampling but also introduces a novel cross-level feature fusion connection, which is a more advanced and effective feature methodology. At the same time, the learnable weights will be allocated to each fused feature, enabling the adjustment of the contributions from various features and enhancing the utilization of crucial feature layers. According to [36], the weighted fusion method is used in this work, and the feature fusion results O_n can be expressed as follows:

$$O_n = \sum_{i=0}^{2} \frac{\lambda_n^i}{\sum_{j=0}^{2} \lambda_n^j + \varepsilon} \cdot I_i \ , \ n \in \{3,4\} \tag{1}$$

where n represents different nodes of the BiFPN-like structure. I_i and λ_n^i denote the input at level i and the learnable weight parameter of level i, respectively. Moreover, ε represents a small constant employed to maintain the stability of the formula.

As shown in Figure 5, inspired by the traditional BiFPN structure, a BiFPN-like feature fusion structure is used in our proposed YOLOv7-UAV algorithm. We employ the extracted features after 4 and 8 times down-sampling to execute a weighted cross-level connection. The more advanced connection method compared to PANnet can better fuse high-level semantic information and low-level geometric information, thereby enhancing the accuracy of object detection in UAV images. In addition, the structure of edge nodes from C2 is retained, as low-level geometric feature information is highly advantageous for recognizing small objects.

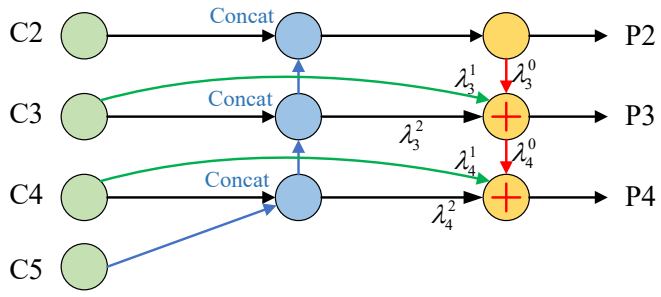

Figure 5. The BiFPN-like structure of YOLOv7-UAV.

3.4. Regression Loss SIoU

The overall loss (L_{total}) in the traditional YOLOv7 algorithm is defined by a weighted loss equation [24]:

$$L_{\text{total}} = W_1 \times L_{\text{box}} + W_2 \times L_{\text{obj}} + W_3 \times L_{\text{cls}} \tag{2}$$

where L_{box}, L_{obj}, and L_{cls} denote the localization loss, confidence loss, and classification loss, respectively. W_1, W_2, and W_3 represent the weight values assigned to these parameters, respectively.

The CIoU function is responsible for calculating the coordinate regression loss function for the traditional YOLOv7 algorithm, which incorporates considerations of the overlap area, the length–width ratio and the distance between the ground-truth box and the prediction box [37]. In addition to the three factors mentioned above, the SIoU loss function includes an angle loss to speed up the training convergence process [38]. The SIoU regression loss function can be expressed as follows:

$$L_{\text{box}} = 1 - \text{IoU} + \frac{\Delta + \Omega}{2} \tag{3}$$

where Ω, IoU, and Δ denote the shape loss, intersection-over-union loss, and distance loss, respectively.

The IoU is calculated as the ratio of the intersection and union of the ground-truth box and prediction box, denoted by [38]

$$\text{IoU} = \frac{B \bigcap B^{gt}}{B \bigcup B^{gt}} \tag{4}$$

where B^{gt} and B denote the ground-truth box and the prediction box, respectively. The $\bigcap$ and $\bigcup$ operations are the intersection and union of the two boxes, respectively.

The shape cost Ω considers the relationship between the prediction box and the ground-truth box and is given by [38]

$$\Omega = \left(1 - e^{-\frac{|w - w^{gt}|}{\max(w, w^{gt})}}\right)^{\theta} + \left(1 - e^{-\frac{|h - h^{gt}|}{\max(h, h^{gt})}}\right)^{\theta} \tag{5}$$

where w and w^{gt} denote the width of the prediction box and the ground-truth box, respectively. h and h^{gt} denote the heights of the prediction box and the ground-truth box, respectively. Moreover, θ is an adjustable variable representing the weight of the network on the shape cost.

The distance cost Δ corresponds to the distance between the centers of two boxes, which is given by [38]

$$\Delta = 2 - e^{(\Lambda - 2) \times \left(\frac{C_{\tilde{y}}}{C_y}\right)^2} - e^{(\Lambda - 2) \times \left(\frac{C_{\tilde{x}}}{C_x}\right)^2} \tag{6}$$

where $C_{\tilde{y}}$ and $C_{\tilde{x}}$ denote the height and width of the rectangle constructed diagonally by connecting the center points of the ground-truth box and the prediction box, respectively. C_y and C_x denote the height and width of the minimum bounding rectangle of the two boxes, respectively.

As shown in Figure 6, the angle loss Λ is introduced as a new factor in the SIoU regression loss function, which is expressed as follows [38]:

$$\Lambda = 1 - 2\sin^2\left(\arcsin(x) - \frac{\pi}{4}\right) \tag{7}$$

$$x = \frac{C_{\tilde{y}}}{\sigma} = \sin(\alpha) \tag{8}$$

where σ represents the distance between the centers of the ground-truth box and the prediction box. α and β denote the included angles between the line connecting the centers of the two boxes and the x-axis and y-axis, respectively.

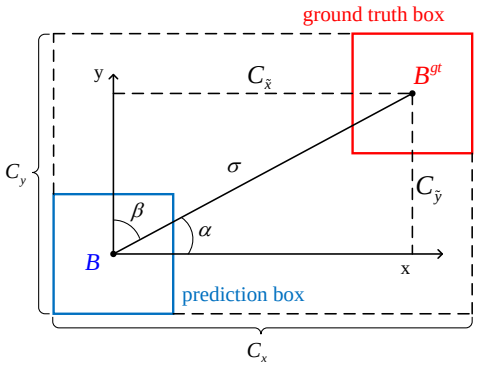

Figure 6. Angle cost of SIoU.

The SIoU function has the capability to adjust the position of the prediction box toward the nearest axis by minimizing the angle α or β: i.e., α is minimized if $\alpha \leq 45°$, and otherwise, β is minimized [38].

4. FPGA Implementation

In the deployment of neural networks, the Vitis-AI scheme has been proven to be the most efficient strategy [39]. This preference for a prolonged development process is inherent in the existing register-transfer-level strategy and the redundancy during the conversion process of the high-level-synthesis strategy. In contrast, Vitis-AI simplifies the deployment of neural networks on FPGA, alleviating deployment complexities. This section elaborates on the deployment of YOLOv7-UAV on FPGA using the Vitis-AI scheme.

4.1. Vitis-AI Overview

Xilinx has introduced the Vitis-AI architecture to enable the deployment of edge-accelerated applications for software or algorithms. It offers an efficient deployment process for edge applications and includes optimization tools, libraries, and sample models. The Vitis-AI-based neural network design flow is shown as follows:

(a) Model Construction: Vitis-AI supports trained network models via popular deep learning frameworks, or the official pre-trained models provided by Xilinx (San Jose, CA, USA) can be chosen.

(b) Development Tools: The quantizer, compiler, and optimizer provided by Vitis-AI allow the processing of a network, resulting in the generation of executable files for hardware.

(c) Overlay: The deep learning processing unit (DPU) can be integrated as an intellectual property (IP) core on FPGA, which is a programmable engine optimized to accelerate the inference of deep learning models and supports neural network architectures such as convolution, an average pooling layer, a maximum pooling layer, and a fully connected layer. Vitis-AI can offer a number of different DPUs for the Xilinx platform, such as the DPUCZDX8G, which is designed for the Zynq UltraScale+ MPSoC. The DPUCZDX8G can be designed for a variety of architectures to meet the resource requirements of different hardware platforms, and larger architectures can provide higher performance with more resources [40].

4.2. Model Quantization and Compilation

As shown in Figure 7, in order to reduce the resource consumption of the model and deployment by Vitis-AI, the proposed YOLOv7-UAV algorithm is quantified and compiled by the AI quantizer and AI compiler based on Vitis-AI.

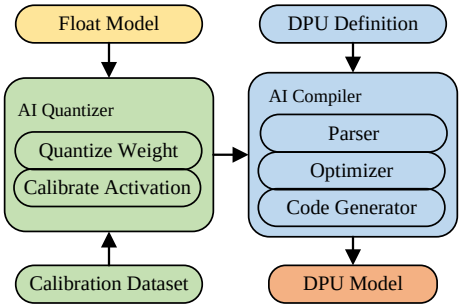

Figure 7. Model quantization and compilation process of Vitis-AI.

The quantization uses a low-bit fixed-point format instead of a high-bit floating point, which can reduce the consumption of hardware resources. In addition, FPGA is not good at handling floating-point type data. Common network models are composed of 32-bit floating-point types. However, in this paper, we reduce the parameters of the network by quantizing them to the 8-bit fixed-point type.

The process of compilation involves generating a sequence of DPU instructions from the DPU definition file and the model file. During this compilation process, to convert the model file into a DPU kernel file that can be used by the DPU IP core, the Vitis-AI compiler maps the network model into an efficient instruction set and data stream that can reuse as much on-chip memory as possible. The Vitis-AI compiler then generates the computational instruction codes and register files that control the computation of the DPU, and each high-level node in the traditional input neural network computation graph is translated into one or more instructions.

4.3. Hardware Platform Processing

The YOLOv7-UAV model will be deployed on the Zynq hardware platform after quantization and compilation into the DPU executable file. The Zynq device includes a Processing System (PS) that is based on the advanced RISC machine (ARM) and programmable logic (PL) that is based on FPGA [41]. First, the DPU IP will be carried in the hardware platform file in the PS part for neural network acceleration. The architecture and configuration of the DPU IP core must take into account the hardware resources of FPGA. Second, the Linux system is built based on the hardware platform file and includes some necessary libraries. Finally, the Linux system will be deployed in the PS part to run the DPU executable file by the Vitis-AI Runtime.

The hardware platform processing flow is shown in Figure 8. First, the UAV image is sent to the input of the PS part and subjected to a sequence of pre-processing procedures, such as image decoding. Second, the image is sent to the DPU IP core in the PL part for the inference process of YOLOv7-UAV, where the network detects the locations and classes of objects. Finally, the results are output to the PS part for drawing and displaying the prediction.

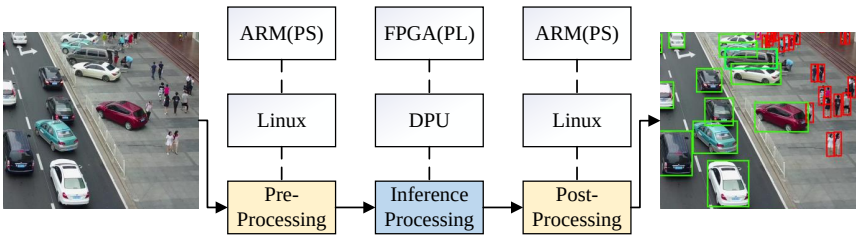

Figure 8. Processing flow of proposed YOLOV7-UAV.

5. Experiments

The VisDrone2019 dataset was employed for both training and evaluating the YOLOv7-UAV algorithm [33]. This dataset considers common objects in UAV images, such as

cars, buses, trucks, and people. Our proposed YOLOv7-UAV algorithm was deployed to compare the performance of the FPGA platform and the GPU platform based on frame per second (FPS) and power.

5.1. Training Details

We implemented the YOLOv7-UAV algorithm in Keras and TensorFlow. All of our models used a Tesla V100 GPU for training and testing. The adam optimizer was used to train the models, and we adopted 0.001 as the initial learning rate with the cosine annealing algorithm.

5.2. Ablation Study

As shown in Table 2, the ablation experiments were completed on VisDrone2019 to validate the effects of the main methods in YOLOv7-UAV. The first row shows the performance of the traditional YOLOv7. From the second row to the last row, mAP@0.5 gradually increases from 41.4% to 45.3%, and the parameters decrease to 26.77M.

Table 2. Ablation study on VisDrone2019-DET-VAL dataset.

Method	mAP@0.5 (%)	Parameters (M)
YOLOv7	41.40	37.29
YOLOv7 + removeP5	43.24 (↑1.84)	26.21
YOLOv7 + removeP5 + P2	44.13 (↑0.89)	26.77
YOLOv7 + removeP5 + P2 + SIoU	44.70 (↑0.57)	26.77
YOLOv7 + removeP5 + P2 + SIoU + BiFPN-like	45.30 (↑0.60)	26.77

The effect of removing the P5 prediction head. Removing the deep-level P5 from the traditional structure of YOLOv7 reduces the number of layers and parameters in the network. The accuracy of the model is improved by 1.84% due to a reduction in the effect of excessive down-sampling on small object detection, demonstrating the effectiveness of removing the P5 prediction head.

The effect of adding the P2 prediction head. Adding the extra prediction head increases the number of layers and model parameters, but the P5 prediction head that we removed is deeper, so it actually reduces the total model parameters. P2 makes the detector more sensitive to small objects, and the increased mean average precision demonstrates the effectiveness of the additional low-level prediction head.

The effect of using the BiFPN-like structure. The BiFPN-like structure uses cross-level connections with weight parameters to enhance the fusion performance of multi-scale features, and it retains edge nodes that contain rich feature information of small objects. The results demonstrate the validity of the BiFPN-like structure compared to the PANnet structure.

The effect of using the SIoU loss function. Incorporating the angle cost into the SIoU loss function facilitates a more efficient adjustment of the prediction box toward the ground-truth box compared to the traditional CIoU loss function. As shown in Table 2, this new SIoU loss function increases accuracy and demonstrates validity.

The effect of the method ensemble. As shown in Table 2, the proposed YOLOv7-UAV algorithm consists of all the above innovative methods, achieving a 3.9% improvement in mAP@0.5 and a 10.52 M reduction in parameters compared to the traditional YOLOv7. The above experiments clearly indicate that the proposed scheme in this paper exhibits advantages in both mAP@0.5 and parameter optimization. Although YOLOv7-UAV adds the prediction head P2 and its feature extraction network, a commensurate reduction in overall system complexity is achieved through the design with the removal of the deep-level prediction head P5. This streamlined configuration renders the algorithm better suited for subsequent deployment on FPGA.

Some detection results on the VisDrone dataset. As shown in Figure 9, we use different colors for the bounding box for different categories and show the corresponding confidence.

The results clearly show that our proposed YOLOv7-UAV algorithm is more effective in detecting small objects, and it can find more objects and has a higher confidence level.

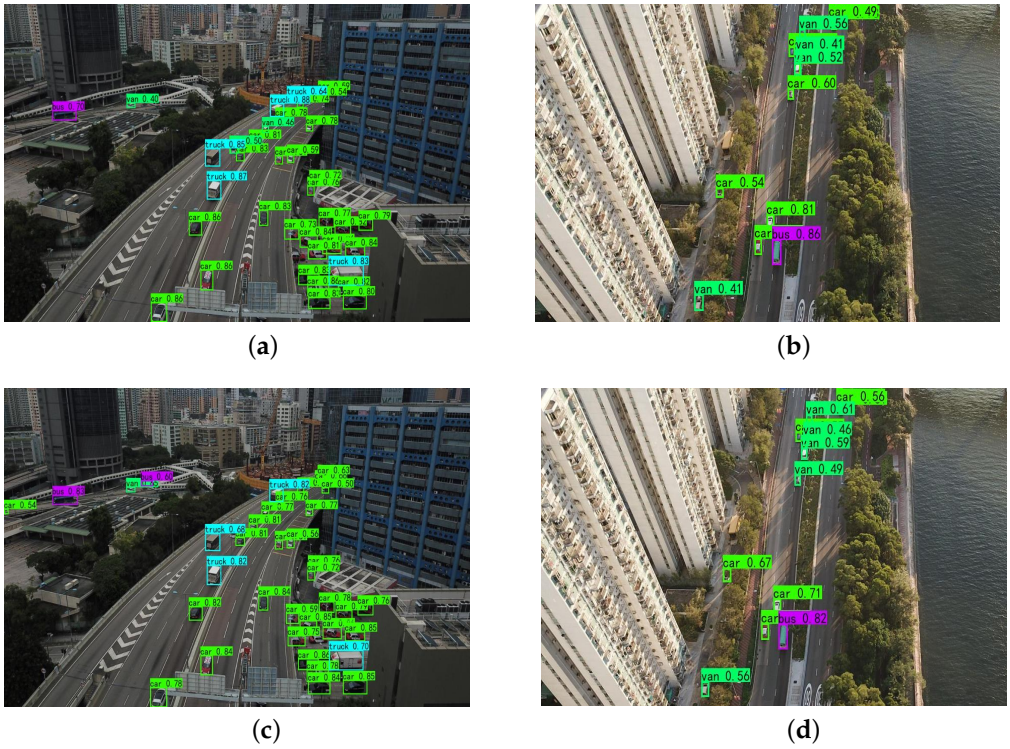

Figure 9. Some visualization results. (**a**,**b**) Results from YOLOv7. (**c**,**d**) Results from YOLOv7-UAV.

5.3. Comparative Experiment

At the same time, we also conducted experiments with YOLOv3, YOLOv4, YOLOv5, and YOLOv7 on the VisDrone dataset. Five evaluation indicators, mAP@0.5, parameter, model size, GPU utilization, and FPS, were calculated, as shown in Table 3.

Table 3. Comparison of different algorithms.

Algorithm	mAP@0.5 (%)	Parameter (M)	Model Size (MB)	GPU Utilization (%)	FPS
YOLOv3	29.41	61.62	235	83	32
YOLOv4	29.68	64.05	244	86	28
YOLOv5	34.53	46.25	176	78	35
YOLOv7	41.40	37.29	142	79	33
YOLOv7-UAV	45.30	26.77	102	79	30

Compared to the four object detection algorithms mentioned above, the mAP@0.5, parameter, and model size evaluation results of the proposed YOLOv7-UAV are the best. In particular, our proposed algorithm outperforms the others in terms of both accuracy and memory consumption. The mAP@0.5 of YOLO-UAV is as high as 45.30%, which is greater than that of the other four algorithms: i.e., it has 15.89% 15.89%, 15.62%, 10.77%, and 3.9% performance gains compared to YOLOv3, YOLOv4, YOLOv5, and YOLOv7, respectively. In particular, the reduced memory consumption is beneficial for deployment with the limited hardware resources of UAVs.

5.4. FPGA Implementation Results

The hardware platform is the ALINX AXU3EG based on the Xilinx Zynq UltraScale+ MPSoC development platform. To deploy the YOLOv7-UAV algorithm on it, we made the following preparations. First, the hardware project containing the DPU IP core was built in Vivado software, and the project was converted to a bitstream file [42]. B1600 was chosen as the system architecture of the DPU, which mainly considers the hardware

resources of the AXU3EG. Table 4 shows the detailed information of these parameters. The usage of the lookup table (LUT), flip-flops (FFs), blocked random-access memory (BRAM), and digital signal processing (DSP) is 62.60%, 46.13%, 75.93%, and 86.67%, respectively. Second, a Linux system was created based on the bitstream file in Petalinux software, and the OpenCV library was added to pre-process and post-process the images [43]. Finally, the DPU executable model files of the YOLOv7-UAV algorithm and Linux system were copied to the SD card, and the whole system was run by the SD card.

Table 4. Hardware resource utilization of FPGA.

Resource	Utilization	Available	Percentage Utilization (%)
LUT	44,169	70,560	62.60
FF	65,099	141,120	46.13
BRAM	164	216	75.93
DSP	312	360	86.67

In order to deploy the YOLOv7-UAV model on the FPGA platform, we used the quantization operation based on Vitis-AI while using a calibrated dataset to reduce the loss of accuracy. Table 5 shows the comparison between the FPGA (AXU3EG) platform, the GPU (Tesla V100) platform, and the embedded AI (Jetson TX2) platform. The AXU3EG platform can achieve 22 FPS when the input image size is 640 × 640. Although the FPS of Tesla V100 is higher, the power consumption of Tesla V100 is up to 250 W, which is 17 times more than the power consumption of AXU3EG. The huge power consumption of the GPU cannot meet the low-power requirements of UAVs. If the energy efficiency of AXU3EG is divided by the energy efficiency of Tesla V100, it is found that AXU3EG is about 12 times more energy-efficient than Tesla V100. Although AXU3EG and Jetson TX2 have the same low power consumption level, the FPS of AXU3EG is almost ten times higher than that of Jetson TX2. Compared to Jetson TX2, the AXU3EG deployment performs faster inference and is more suitable for UAV image object detection tasks with a real-time requirement. Therefore, the FPGA-based object detection deployment platform designed in our study is more suitable for deployment on low-power UAVs and can be extended to similar edge devices with limited energy.

Table 5. Performance on different hardware platforms.

Platform	Input	FPS	Power (W)	Energy Efficiency (FPS/W)
Tesla V100	640 × 640	30	250	0.12
Jetson TX2	640 × 640	2	15	0.13
AXU3EG	640 × 640	22	15	1.46
Tesla V100	416 × 416	59	250	0.23
Jetson TX2	416 × 416	4	15	0.26
AXU3EG	416 × 416	43	15	2.86

We also compared other algorithms on the same FPGA (AXU3EG) platform to clearly understand the advantages and disadvantages of the proposed methods, as shown in Table 6. YOLOv3, YOLOv4, YOLOv5, and YOLOv7 were also implemented on FPGA with the described Vitis-AI scheme. The FPS of our proposed YOLOv7-UAV algorithm appears to have decreased from 33 to 22. This may be due to the large scale of features in the added prediction head (P2) and its associated networks, resulting in extended computation times during the DPU inference on FPGA. Nevertheless, the FPS of our proposed YOLOv7-UAV is still at an acceptable level for real-time applications on UAVs.

Table 6. Comparison of different algorithms on FPGA.

Algorithm	Input	FPS	Power (W)	Energy Efficiency (FPS/W)
YOLOv3	640 × 640	28	15	1.86
YOLOv4	640 × 640	28	15	1.86
YOLOv5	640 × 640	30	15	2
YOLOv7	640 × 640	33	15	2.2
YOLOv7-UAV	640 × 640	22	15	1.46

6. Conclusions

UAV images are characterized by a large proportion of small objects and dense object regions. In response to these challenges, this paper proposes an efficient YOLOv7-UAV algorithm to detect small objects in UAV image scenarios. YOLOv7-UAV adds an extra low-level prediction head (P2) to extract the detailed information of small objects, removes a deep-level prediction head (P5) to reduce the effect of multiple down-sampling operations, employs the BiFPN-like weighted cross-level connection structure to improve the fusion performance of multi-scale feature maps, and adopts the SIoU loss function as the regression loss function to improve the convergence efficiency, which improves the algorithm's ability to detect small object features. The final experimental results demonstrate that our proposed YOLOv7-UAV algorithm is highly applicable in UAV image scenarios, achieving a mAP of 45.3% on the VisDrone2019 dataset. Finally, YOLOv7-UAV is quantified and compiled by Vitis-AI and deployed on the FPGA (AXU3EG) platform to achieve higher energy efficiency, which is improved by 12 times compared to the GPU platform. The proposed YOLOv7-UAV algorithm in this paper is designed for the specific application of UAV image scenarios, but the additional P2 prediction head and the removed P5 prediction head focus on detecting small objects, which can be extended to other small object detection domains in the future.

Author Contributions: Conceptualization, X.L.; methodology, X.L. and Y.W.; software, Y.W.; validation, J.L. and W.D.; investigation, J.L., W.D. and Y.W; resources, X.Z. and Y.H.; writing—original draft preparation, X.L. and Y.W; writing—review and editing, Y.W., X.Z. and Y.H. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by the National Natural Science Foundation of China under Grant 62201479, in part by the Natural Science Foundation of Sichuan Province under Grant 2023NSFSC1388, in part by the National Natural Science Foundation of China under Grant 62101386, in part by the Shanghai Sailing Program under Grant 21YF1450000, in part by the Fundamental Research Funds for the Central Universities under Grant 22120230311, in part by the Guangdong Provincial Key Laboratory of Future Networks of Intelligence, the Chinese University of Hong Kong, Shenzhen, under Grant 2022B1212010001-OF04, and in part by the Key Laboratory of Medicinal and Edible Plant Resources Development of Sichuan Education Department, the Chengdu University, under Grant 10Y202201.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Publicly available datasets were analyzed in this study. This data can be found here: http://aiskyeye.com/ (accessed on 3 January 2024).

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

References

1. Zou, Z.X.; Chen, K.Y.; Shi, Z.W.; Guo, Y.H.; Ye, J.P. Object detection in 20 years: A survey. *Proc. IEEE* **2023**, *111*, 257–276. [CrossRef]
2. Zaidi, S.S.A.; Ansari, M.S.; Aslam, A.; Kanwal, N.; Asghar, M.; Lee, B. A survey of modern deep learning-based object detection models. *Digit. Signal Process* **2022**, *126*, 103514. [CrossRef]
3. Ghahremannezhad, H.; Shi, H.; Liu, C.J. Object detection in traffic videos: A survey. *IEEE Trans. Intell. Transp. Syst.* **2023**, *24*, 6780–6799. [CrossRef]
4. Wang, Y.; Sun, Q.Y.; Liu, Z.Z.; Gu, L. Visual detection and tracking algorithms for minimally invasive surgical instruments: A comprehensive review of the state-of-the-art. *Robot. Auton. Syst.* **2022**, *149*, 103945. [CrossRef]

5. Zhong, L.T.; Zhang, X.Q.; Ran, L.Y.; Han, Y.M.; Chu, H.Y. Visual SLAM for dynamic environments based on static key-points detection. In Proceedings of the International Conference on Virtual Reality (ICVR), Xianyang, China, 12–14 May 2023.
6. Zhou, Z.Y.; Zhang, X.Q.; Ran, L.Y.; Han, Y.M.; Chu, H.Y. DSC-GraspNet: A lightweight convolutional neural network for robotic grasp detection. In Proceedings of the International Conference on Virtual Reality (ICVR), Xianyang, China, 12–14 May 2023.
7. Li, X.M.; Xu, J. Positioning optimization for sum-rate maximization in UAV-enabled interference channel. *IEEE Signal Process. Lett.* **2019**, *26*, 1466–1470. [CrossRef]
8. Heidari, A.; Navimipour, N.J.; Unal, M.; Zhang, G.D. Machine learning applications in internet-of-drones: Systematic review, recent deployments, and open issues. *ACM Comput. Surv.* **2022**, *55*, 1–45. [CrossRef]
9. Wu, X.; Li, W.; Hong, D.F.; Tao, R.; Du, Q. Deep learning for unmanned aerial vehicle-based object detection and tracking: A survey. *IEEE Geosci. Remote. Sens. Mag.* **2022**, *10*, 91–124. [CrossRef]
10. Zhu, P.; Wen, L.; Du, D.; Bian, X.; Fan, H.; Hu, Q.; Ling, H. Detection and tracking meet drones challenge. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *44*, 7380–7399. [CrossRef]
11. 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.
12. Wang, C.-Y.; Bochkovskiy, A.; Liao, H.-Y.M. Scaled-YOLOv4: Scaling cross stage partial network. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 19–25 June 2021.
13. Xie, X.Z.; Lu, G. A research of object detection on UAVs aerial images. In Proceedings of the International Conference on Big Data & Artificial Intelligence & Software Engineering (ICBASE), Zhuhai, China, 24–26 September 2021.
14. Li, X.M.; Yin, B.L.; Wei, L.L.; Zhang, X.Q. Reinforcement learning-based age of information optimization in UAV-enabled communication system. *J. Univ. Electron. Sci. Technol. China* **2022**, *51*, 213–218.
15. Wang, L.; Zhou, H.; Bian, C.J.; Jiang, K.; Cheng, X.L. Hardware acceleration and implementation of YOLOX-s for on-orbit FPGA. *Electronic* **2022**, *11*, 3473. [CrossRef]
16. Wei, Y.K.; Li, J.H.; Duan, W.W.; Li, X.M.; Zhang, X.Q.; Huang, Y. YOLOv7-UAV: Improved YOLOv7 algorithm for small object detection in UAV image scenarios. In Proceedings of the International Conference on Artificial Intelligence of Things and Systems (AIoTSys), Xi'an, China, 19–22 October 2023.
17. Sirisha, U.; Praveen S.P.; Srinivasu, P.N.; Barsocchi, P.; Bhoi, A.K. Statistical analysis of design aspects of various YOLO-based deep learning models for object detection. *Int. J. Comput. Intell. Syst.* **2023**, *16*, 126. [CrossRef]
18. Girshick, R. Fast R-CNN. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 13–16 December 2015.
19. He, K.M.; Gkioxari, G.; Dollar, P.; Girshick, R. Mask R-CNN. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *42*, 386–397. [CrossRef]
20. Sikora, P.; Malina, L.; Kiac, M.; Martinasek, Z.; Riha, K.; Prinosil, J.; Jirik, L.; Srivastava, G. Artificial intelligence-based surveillance system for railway crossing traffic. *IEEE Sens. J.* **2021**, *21*, 1551–15526. [CrossRef]
21. 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 (CVPR), Las Vegas, TX, USA, 26–31 June 2016.
22. Ganesh, P.; Chen, Y.; Yang, Y.; Chen, D.; Winslett, M. YOLO-ReT: Towards high accuracy real-time object detection on edge GPUs. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), Waikoloa, HI, USA, 3–8 January 2022.
23. Liang, S.; Wu, H.; Zhen, L.; Hua, Q.; Garg, S.; Kaddoum, G.; Hassan, M.M.; Yu, K. Edge YOLO: Real-time intelligent object detection system based on edge-cloud cooperation in autonomous vehicles. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 25345–25360. [CrossRef]
24. 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 (CVPR), Vancouver, BC, Canada, 18–22 June 2023.
25. Zhu, X.K.; Lyu, S.C.; Wang, X.; Zhao, Q. TPH-YOLOv5: Improved YOLOv5 based on transformer prediction head for object detection on drone-captured scenarios. In Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops (ICCVW), Montreal, ON, Canada, 11–17 October 2021.
26. Li, Y.Y.; Huang, Q.; Pei, X.; Chen, Y.Q.; Jiao, L.C.; Shang, R.H. Cross-layer attention network for small object detection in remote sensing imagery. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2021**, *14*, 2148–2161. [CrossRef]
27. Deng, C.F.; Wang, M.M.; Liu, L.; Liu, Y.; Jiang, Y.L. Extended feature pyramid network for small object detection. *IEEE Trans. Multimed.* **2022**, *24*, 1968–1979. [CrossRef]
28. Chen, Z.; Liu, C.; Filaretov, V.F.; Yukhimets, D.A. Multi-scale ship detection algorithm based on YOLOv7 for complex scene SAR images. *Remote Sens.* **2023**, *15*, 2071. [CrossRef]
29. Redmon, J.; Farhadi, A. YOLOv3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
30. Yang, J.L.; Yang, H.; Wang, F.; Chen, X. A modified YOLOv5 for object detection in UAV-captured scenarios. In Proceedings of the IEEE International Conference on Networking, Sensing and Control (ICNSC), Shanghai, China, 15–18 December 2022.
31. Xue, S.; Li, Z.Y.; Wu, R.; Zhu, T.T.; Yuan, Y.C.; Ni, C. Few-shot learning for small impurities in tobacco stems with improved YOLOv7. *IEEE Access* **2023**, *11*, 48136–48144. [CrossRef]
32. Arthur, D.; Vassilvitskii, S. K-means++: The advantages of careful seeding. In Proceedings of the Annual ACM-SIAM Symposium on Discrete Algorithms, New Orleans, LA, USA, 7–9 January 2007.

33. Du, D.; Zhu, P.; Wen, L.; Bian, X.; Lin, H.; Hu, Q.; Peng, T.; Zheng, J.; Wang, X.; Zhang, Y.; et al. VisDrone-DET2019: The vision meets drone object detection in image challenge results. In Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops (ICCVW), Seoul, Republic of Korea, 27–28 October 2019.
34. Lin, T.-Y.; Dollar, P.; Girshick, R.; He, K.M.; Hariharan, B.; Belongie, S. Feature pyramid networks for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017.
35. Liu, S.; Qi, L.; Qin, H.F.; Shi, J.P.; Jia, J.Y. Path aggregation network for instance segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–22 June 2018.
36. Tan, M.X.; Pang, R.M.; Le, Q.V. EfficientDet: Scalable and efficient object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 14–19 June 2020.
37. Zheng, Z.; Wang, P.; Ren, D.; Liu, W.; Ye, R.; Hu, Q.; Zuo, W. Enhancing geometric factors in model learning and inference for object detection and instance Segmentation. *IEEE Trans. Cybern.* **2022**, *52*, 8574–8586. [CrossRef] [PubMed]
38. Gevorgyan, Z. SIoU loss: More powerful learning for bounding box regression. *arXiv* **2022**, arXiv:2205.12740.
39. Xilinx. Vitis AI User Guide (UG1414). Available online: https://docs.xilinx.com/r/2.0-English/ug1414-vitis-ai (accessed on 27 December 2023).
40. Xilinx. DPUCZDX8G for Zynq UltraScale+ MPSoCs Product Guide (PG338). Available online: https://docs.xilinx.com/r/4.0-English/pg338-dpu (accessed on 27 December 2023).
41. Xilinx. Zynq UltraScale+ MPSoC Data Sheet: Overview (DS891). Available online: https://docs.xilinx.com/v/u/en-US/ds891-zynq-ultrascale-plus-overview (accessed on 27 December 2023).
42. Xilinx. Vivado Design Suite User Guide: Getting Started (UG910). Available online: https://docs.xilinx.com/r/en-US/ug910-vivado-getting-started (accessed on 27 December 2023).
43. Xilinx. Petalinux Tools Documentation: Reference Guide (UG1144). Available online: https://docs.xilinx.com/r/en-US/ug1144-petalinux-tools-reference-guide (accessed on 27 December 2023).