electronics

Article

Wildlife Object Detection Method Applying Segmentation Gradient Flow and Feature Dimensionality Reduction

Mingyu Zhang, Fei Gao *, Wuping Yang and Haoran Zhang

School of Science, Wuhan University of Technology, Wuhan 430070, China
* Correspondence: gaof@whut.edu.cn; Tel.: +86-18971097697

Abstract: This work suggests an enhanced natural environment animal detection algorithm based on YOLOv5s to address the issues of low detection accuracy and sluggish detection speed when automatically detecting and classifying large animals in natural environments. To increase the detection speed of the model, the algorithm first enhances the SPP by switching the parallel connection of the original maximum pooling layer for a series connection. It then expands the model's receptive field using the dataset from this paper to enhance the feature fusion network by stacking the feature pyramid network structure as a whole; secondly, it introduces the GSConv module, which combines standard convolution, depth-separable convolution, and hybrid channels to reduce network parameters and computation, making the model lightweight and easier to deploy to endpoints. At the same time, GS bottleneck is used to replace the Bottleneck module in C3, which divides the input feature map into two channels and assigns different weights to them. The two channels are combined and connected in accordance with the number of channels, which enhances the model's ability to express non-linear functions and resolves the gradient disappearance issue. Wildlife images are obtained from the OpenImages public dataset and real-life shots. The experimental results show that the improved YOLOv5s algorithm proposed in this paper reduces the computational effort of the model compared to the original algorithm, while also providing an improvement in both detection accuracy and speed, and it can be well applied to the real-time detection of animals in natural environments.

Keywords: animal recognition; feature fusion networks; YOLOv5s; segmentation gradient flow; GSConv

Citation: Zhang, M.; Gao, F.; Yang, W.; Zhang, H. Wildlife Object Detection Method Applying Segmentation Gradient Flow and Feature Dimensionality Reduction. *Electronics* **2023**, *12*, 377. https://doi.org/10.3390/electronics12020377

Academic Editor: Silvia Liberata Ullo

Received: 26 November 2022
Revised: 5 January 2023
Accepted: 6 January 2023
Published: 11 January 2023
Corrected: 19 April 2023

1. Introduction

Target identification and recognition of animals have grown in importance as computer vision technology has progressed. However, conventional approaches to these problems currently do not produce satisfying outcomes, and deep learning has emerged as a breakthrough technology in this area. In recent centuries, the expansion of human society into the natural environment for development has resulted in the loss of wildlife habitats, and the environment has been severely damaged by the advent of the industrial age and rapid population growth. Some fauna have already become extinct as a result of these. Therefore, a novel method for wildlife conservation and ecological study is provided by the application of target detection algorithms in deep learning to detect and identify animals [1].

Convolutional neural networks (CNN) are a class of feedforward neural networks (FNN) with convolutional computation and a deep structure, which is one of the representative algorithms of deep learning [2]. With the development of artificial intelligence and deep learning, the application of convolutional neural networks to wildlife detection and identification is of great significance for wildlife conservation as it extracts surrounding target features in real time. Among the algorithms for target feature extraction, the faster region-based convolutional neural network (Faster R-CNN) algorithm [3], single shot multibox detector (SSD) algorithm [4,5], and the you only look once (YOLO) algorithm [6–8] have successfully applied deep learning to target extraction and target detection; the YOLO algorithm is trained and detected in a separate network, and regression and classification

are performed directly on the whole graph in the CNN, so its performance is improved compared to the Faster R-CNN algorithm and the SSD algorithm, but its recognition accuracy for small or distant targets is poor [9,10].

In 2020, YOLOv5 was proposed, and its basic structure was divided into the Backbone, Neck, and Head. YOLOv5s is a subset of YOLOv5, and numerous researchers have made great strides in fusing the target detection algorithm with real-world uses. Jiale Yao et al. solved the recognition of vehicle targets in bad weather by increasing the number of model parameters, merging Transformer and CBAM into the YOLOv5 algorithm, and optimizing the parameters of the Backbone of YOLOv5 algorithm, using the loss function of EIOU instead of the original loss function of CIOU, which is beneficial for the recognition of vehicles [11]. Hao Wang and Shixin Sun et al. created a reinforcement-learning-based system for improving underwater images, which is comparable to target recognition in bad weather. YOLOv5 is a lightweight, quick, and accurate object detection method for underwater environments according to preliminary testing findings. They used a Markov decision process (MDP) to describe the improvement of underwater images. The MDP can represent a variety of improved outcomes for underwater photographs after being trained with reinforcement learning. Their reinforcement learning architecture provided a series of actual actions that are transparent from an implementation standpoint, in contrast to the black-box processing approach of deep learning methods. The outcomes of the experiments supported the framework for reinforcement learning's efficacy in improving underwater image quality [12,13]. This has similarities to the identification of animals in different environments which follows in this paper.

Weimin Liu et al. used coordinate attention to improve YOLOv5 to reduce the loss of feature information and reduced its size by the lightweight method ShuffleNetV2 [14]. Fenghua Wang et al. used Ghostconv to replace the convolutional layer in YOLOv5s CSP to improve the detection speed by lightweight network structure and then, as in the above paper, introduced the BiFPN module to improve the PANet structure of the Neck to improve the detection accuracy of Xiaomila green pepper in surroundings similar to the target [15]. In the meantime, other researchers have made progress by fusing target detection algorithms with animal recognition applications. Ramakant Chandrakar et al. presented a system for automatic detection and recognition of animals using deep CNN with genetic segmentation for animal detection [16]. For image fusion, S. Divya Meena et al. proposed a dual-scale image decomposition-based fusion technique (DDF) that fuses visible and thermal images and introduced a seed-labels-focused object detector (SLOD) [17].

The proposed networks were applied to edge devices; in addition to YOLOv5s, Jiadong Chen et al. proposed convolution kernel first (CKF), an efficient scheme for designing memristor-based fully convolutional neural networks (FCNs). The parameters and circuit power consumption of the edge device are both reduced by CKF. The test set maintains high accuracy while lowering power loss, as shown by the simulation results of real medical image segmentation tasks [18]. Bo Lyu et al. proposed the deployment of spectral graph convolutional networks (GCNs) on memristive crossbars. They also provided an accelerated technique that combines diagonal block matrix multiplication with sparse Laplace matrix reordering. The results showed that the method was effective when used in the supervised learning graph dataset (QM7) and unsupervised learning dataset (karate club). The outcomes showed that the model maintained a high level of accuracy and achieved a memristor number reduction, which is crucial for future network deployment on edge devices [19]. Subeen Lee et al. introduced the task discrepancy maximization (TDM) module. The support attention module (SAM) and query attention module (QAM) are two novel components that TDM uses to learn task-specific channel weights [20]. Heng Li et al. introduced a gated recurrent unit to improve the result by tracing the temporal information of the cost graph [21]. To address the issue of gradient disappearance, we also present the attention mechanism module in this article while adjusting the weights on each channel and using a normalization unit to combine the number of channels in the model.

However, the use of target detection algorithms to find animals in natural settings is not widely accepted in all respects, and recognition accuracy and speed need to be increased [22]. To improve the accuracy and speed of YOLOv5s in wildlife recognition, this paper replaces the SPP module in the Backbone with the SPPF module to improve the detection speed of the model and adjusts the feature pyramid network structure in the Backbone to enhance the ability of target feature extraction for large sample animals by expanding the sensory field of the model. Secondly, the Conv module in the Head is replaced with the GSConv module to reduce the number of parameters in the model and to enhance the network feature extraction capability. Finally, the VoVGSCSP module is introduced to divide the input feature map into two channels, which enhances the nonlinear representation of the model and solves the problem of gradient disappearance. The testing findings demonstrate that the model can more effectively recognize wildlife in natural settings, has a compact footprint, is simple to deploy on mobile terminals, and has a high detection accuracy.

2. YOLOv5 Algorithm

2.1. YOLOv5 Network Architecture

YOLOv5, proposed by Jocher in 2020, can be divided into four models, YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x, according to increasing network size [23]. The four models differ in the depth and width of the network, and the rest of them are the same.

The network structure of YOLOv5 is shown in Figure 1. YOLOv5 mainly consists of four parts: the Input, Backbone, Neck, and Prediction [24,25]. Input is responsible for pre-processing the input images to meet the training requirements. The Backbone, which includes Focus, CBL, CSP, and SPP [26], is the backbone network responsible for providing image feature information. The Neck is the structural layer containing the fused features of the images and passes the feature information to Prediction. Prediction is responsible for providing prediction frames based on the feature information and filtering the detection frames by non-maximal value suppression.

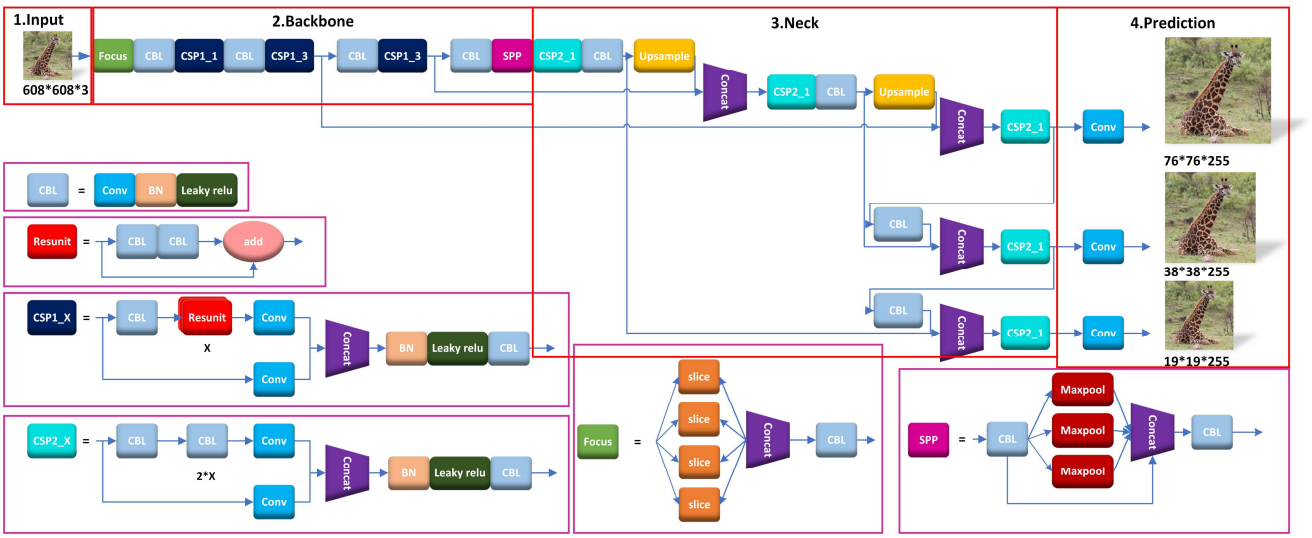

Figure 1. YOLOv5 network structure.

YOLOv5 uses Mosaic data to enrich the dataset at the input side. Four photographs are randomly selected, then they are combined using a random scale and aligning technique. Then, it performs adaptive anchor frame computation to preprocess the images and adaptive scaling to address the black edge problem, which enhances the model's training efficiency and network robustness [27,28]. The Backbone uses a Focus structure for slicing downsampling, which reduces the information entropy brought by convolution. Meanwhile, it improves the CSP structure to C3 by applying C3_1_X and C3_2_X to the Backbone

and Neck, respectively, which enhances the learning ability of the network. Through the use of spatial pyramid pooling (SPP), which can partially address the issue of multi-scale target fusion, it extracts the initial features of the images. The PANet structure used in the Neck consists of a feature pyramid structure of FPN + PAN, with FPN passing top-down information from the higher levels to the lower levels to form the feature map, and PAN passing bottom-up location information to downsample and fuse the feature map. The simultaneous use of both can strengthen the network feature fusion capability, enhance the model's detection function for targets of different sizes, and solve the multi-scale problem. Prediction includes DIoU_NMS and loss function, using CIoU function to calculate the position loss, which solves the problem wherein GIoU degrades to IoU when two target frames intersect. The detection frame is filtered by DIoU_NMS, which can effectively solve the problem of missed detection and improve the accuracy of network prediction [29–33].

2.2. Backbone

2.2.1. CBS

CBS is a composite convolutional module consisting of a convolutional layer, a BN layer, and an activation function layer, which is an important part of many modules. The BN layer mainly normalizes the data and facilitates fast convergence to accelerate the network. The activation function layer uses SiLU as the new activation function, which is essentially a weighted linear combination of the sigmoid. The SiLU function is continuous and smooth. On deeper models, where it can increase the non-linearity of the model and boost detection precision, it performs better than the original activation function LeakyRelu. The expressions are as follows:

$$SiLU(x) = x \cdot sigmoid(x) \tag{1}$$

$$sigmoid(x) = \frac{1}{1 + e^{-x}} \tag{2}$$

2.2.2. C3

C3 is composed of two nested residual modules. The model is simplified, and the number of convolution modules is decreased with this structure without impacting the feature information. Depending on the application location, C3 can be divided into C3_1_X and C3_2_X. C3_1_X is used in the Backbone convolutional neural network part, which contains X residual components (Resunit); the larger the X the deeper the network structure. C3_2_X, on the other hand, is applied to the Neck and contains 2X residual components, the structure of which differs from C3_1_X only in terms of the number of residual components. Increasing the number of Resunit can increase the gradient value of backpropagation between different layers and prevent the gradient degradation problem caused by the deeper structure of the network. Moreover, the model's ability to extract and fuse network features is enhanced. Therefore, along with increasing the network's capacity for learning and lowering its computational and parameter requirements, C3 also increases the network's precision in target detection.

2.2.3. SPP

SPP transforms the input feature map of arbitrary dimension into a fixed dimensional feature vector to ensure the same feature dimension as the fully connected layer. SPP first halves the number of channels by the composite convolution module (CBS), then downsamples it using three maximum poolings of 5×5, 9×9, and 13×13, respectively, and, finally, outputs; thus the output of the convolutional layer retains local features at different scales. The SPP has a skip connection before downsampling, and the three pooling results are overlaid with the image's initial features through the Concat feature connection. This allows the local features to be fused with the global features of the original convolutional layer output, providing good feature extraction capability. Furthermore, the number of channels after feature stacking becomes twice as many as the original, which

increases the number of channels to a larger extent at a smaller cost and improves the field of perception.

2.3. Prediction

DIoU-NMS

NMS is the post-processing of the detection results. It obtains the final detection results by removing the redundant and useless frames for each object. However, NMS tends to filter frames solely based on IoU, which might cause issues when the intersection of frames for various objects is empty.

In YOLOv5, the frames are filtered using DIoU_NMS. Only the frames with the highest scores remain after the NMS filtering because the IoU values are often higher when objects are very close to one another. The DIoU_NMS adds the distance between the midpoints of the frames as an indicator, which effectively solves the problem of missed detections. The calculations are as follows:

$$s_i = \begin{cases} s_i, IoU - R_{DIoU}(M, B_i) < \varepsilon \\ 0, IoU - R_{DIoU}(M, B_i) \geq \varepsilon \end{cases} \tag{3}$$

$$R_{DIoU} = \frac{\rho^2(b, b^{gt})}{c^2} \tag{4}$$

where A is the prediction frame, B is the true frame, and C is the minimum convex set of A and B. Where s_i is the score for the different categories, ε is the threshold set in the NMS; $\rho^2(b, b^{gt})$ is the Euclidean distance between the centroids of A and B; c is the maximum distance of C; i is the number of anchor frames in each grid [34–36].

3. The Improved YOLOv5s Algorithm in This Paper

3.1. SPP Improvements

In this paper, SPPF is improved on SPP, and experiments show that SPPF can achieve the same computational results as SPP, but SPPF is almost twice as fast [37]. As shown in Figure 2, SPPF first halves the number of channels in the feature map using the composite convolution module (CBS) and then downsamples it through the maximum pooling layer, which uses three maximum poolings of size 5×5 in series instead of the three maximum poolings in parallel in SPP to further fuse the image features. It also superimposes the three pooling results with the initial features of the picture to fuse the local features with the global features, changing the number of channels to twice the original at a smaller cost, which improves the receptive field and can solve the problem of multi-scale target fusion to a certain extent. SPPF can convert feature maps of arbitrary dimensions into feature vectors of fixed dimensions and increase the receptive field, which is more efficient than SPP under the condition of having the same adaptive scaling output results [38].

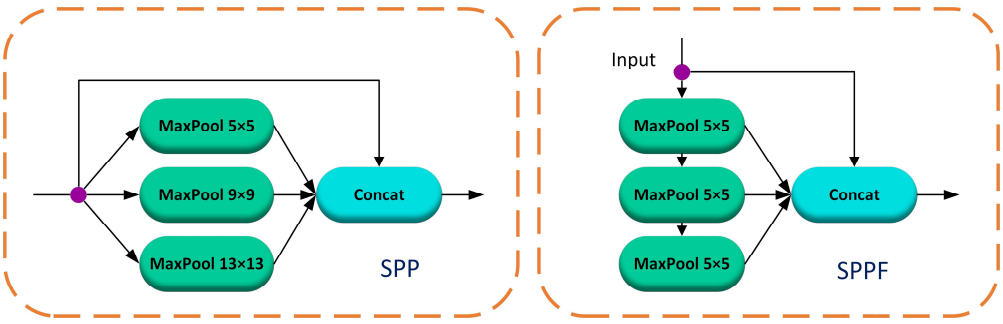

Figure 2. Structure of SPP and SPPF.

3.2. Upgrading of the Feature Pyramid Structure

To enhance the effectiveness of target detection, the receptive field must be expanded due to the large size of the targets in the dataset used in this study. Most studies tend

to increase the receptive field by increasing the convolutional layer and increasing the downsampling ratio [39]. However, in convolutional neural networks, the feature maps obtained by deep convolution are more semantic, but the location information is lost and the computational effort is increased. Therefore, in this paper, we increase the downsampling rate based on the original algorithm [40,41] and stack the feature pyramid network structure one level up, i.e., the original P3–5 structure is improved to a P4–6 structure. The newly added P6 detection layer is more suitable for detecting larger targets and can achieve higher accuracy under higher-resolution training conditions [42].

Due to the small number of downsampling layers of YOLOv5s, the detection effect on large-sized objects is not ideal. Therefore, we add a 64× downsampling feature fusion layer P6 in the Backbone, which is output by the backbone network with 64× downsampling and 1024 output channels, generating a feature map of size 10 × 10. The smaller the feature map, the sparser the newly generated feature map's segmented grid, the more advanced the semantic information contained in every grid, and the larger the receptive field obtained, which is conducive to the recognition of large-sized targets [43]. At the same time, the original 8× downsampling feature fusion layer is removed, i.e., only P4, P5, and P6 are used to downsample the image. In this way, the original image is sent to the feature fusion network after 16×, 32×, and 64× downsampling to obtain 40 × 40, 20 × 20, and 10 × 10 feature maps in the detection layer. Three sizes of feature maps are used to detect targets of different sizes, and the original feature extraction model is shown in Figure 3.

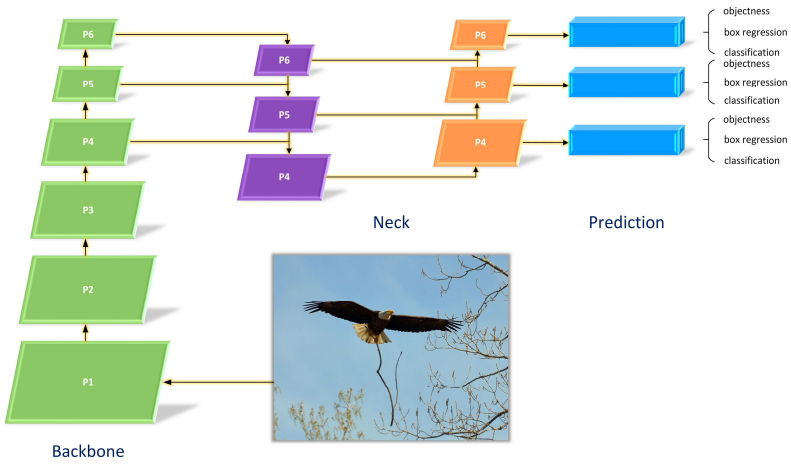

Figure 3. Network structure of the model.

As shown in Figure 3, P4, P5, and P6 are three different layers of feature maps, corresponding to 16, 32, and 64 times downsampling magnification, respectively. Feature maps P4, P5, and P6 carry out feature fusion through feature pyramids, i.e., fusing the high-level and low-level feature maps by passing high-level information from top to bottom and location information from bottom to top, combining the location information of the low-level network with the semantic information of the high-level network. The model can be used to enhance the detection function of targets of different sizes and strengthen the multi-scale prediction capability of the network for targets. P6 has a higher downsampling multiplier and contains a larger receptive field per pixel, which provides more sufficient information on large-sized targets during the fusion of feature information transfer, thus, enhancing the learning capability of the network. The feature map then enters the detection layer for prediction, which consists of three detection heads and is responsible for identifying feature points on the feature map and determining whether there is a target corresponding to it.

We carry out ablation experiments because target detection layers add more parameters. The experimental findings demonstrate that the number of parameters increases only slightly after the P4–P6 structure is improved due to the addition of only feature layers and

not a significant number of extra convolutional layers; however, the detection accuracy is improved. In conclusion, by increasing the downsampling multiplier to obtain a smaller feature map, the feature map receptive field is larger, which is helpful for fully refining the image feature information, reducing the information loss, and strengthening the network's learning capability, thus, improving the accuracy of target detection and recognition while decreasing the computational effort.

3.3. GSConv

Although P4–6 are improved, it still introduces a certain number of parameters, which is not optimal for the creation of lightweight networks even if it leads to significant accuracy advances. The design of lightweight networks often favors the use of depth-wise separable convolution (DSC). The greatest advantage of DSC is its efficient computational power, with approximately one-third of the number of parameters and computational effort of conventional convolution, but the channel information of the input image is separated during the calculation. This deficiency leads to a much lower feature extraction and fusion capability of DSC than even the standard convolution (SC). To make up for this deficiency, MobileNets first compute channel information independently and then fuse it with a large number of dense convolutions; ShuffleNets use shuffle to achieve channel information interaction; GhostNet only inputs half of the number of channels for ordinary convolution to retain the interaction information. Many lightweight networks are limited to similar thinking, but all three approaches use only DSC or SC independently, ignoring the joint role of DSC and SC and, thus, cannot fundamentally solve the problems of DSC [44].

To make effective use of the computational power of DSC and, at the same time, make the detection accuracy of DSC reach the standard of SC, this paper proposes a new hybrid convolutional approach, GSConv, based on research on lightweight networks. The GSConv module is a combination of SC, DSC, and shuffle, and its structure is shown in Figure 4. Firstly, a feature map with the input channel number c_1 is input, half of the channel number is divided for deep separable convolution, and the remaining channel is convolved for normal convolution, after which the two are joined for feature concatenation. Then, the information generated by SC is infiltrated into the various parts of the information generated by DSC using shuffle, and the number of output channels in the feature map is c_2. Shuffle is a channel-mixing technique that was first used in ShuffleNets [45]. It enables channel information interaction by allowing information from the SC to be fully blended into the DSC output by transferring its feature information on various channels.

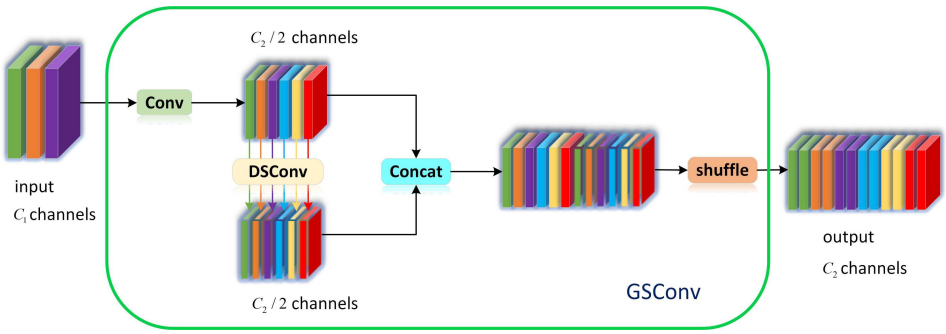

Figure 4. GSConv module structure.

During the convolution process, the spatial information of feature maps is gradually transferred to the number of channels, i.e., the number of channels increases while the width and height of the feature map decrease, thus, making the semantic information stronger and stronger. In contrast, each spatial compression and channel expansion of the feature map results in a partial loss of semantic information, which affects the accuracy of target detection. SC retains the hidden connections between each channel to a greater extent, which can reduce the loss of information to a certain extent, but the time complexity

is greater; on the contrary, DSC completely cuts off these connections, causing the channel information of the input image to be completely separated during the calculation process, that is, the feature map is separable with minimal time complexity. GSConv retains as many of these connections as possible while keeping the time complexity small, which reduces information loss and enables faster operation, achieving a degree of unity between SC and DSC.

The time complexity of the convolution calculation is usually defined by FLOPs, and the time complexity of SC, DSC, and GSConv is calculated as follows:

$$T_{SC} \sim O(WHK_1K_2C_1C_2) \tag{5}$$

$$T_{DSC} \sim O(WHK_1K_2C_2) \tag{6}$$

$$T_{GSConv} \sim O(WHK_1K_2C_2(C_1+1)/2) \tag{7}$$

where W and H are the width and height of the output feature map, respectively; K_1 and K_2 are the size of the convolution kernel; C_1 is the number of channels of the input feature map; C_2 is the number of channels of the output feature map.

Applying each of the three convolution patterns to the same image of the dataset in this paper, the visualization results for SC, DSC, and GSConv are as shown in Figure 5. Compared to DSC, the feature maps output by GSConv are more similar to those output by SC, and, in some cases, the detection of the target is even better than SC with the highest detection accuracy; some of the output colors of DSC are darker, and there is a lack of detection accuracy.

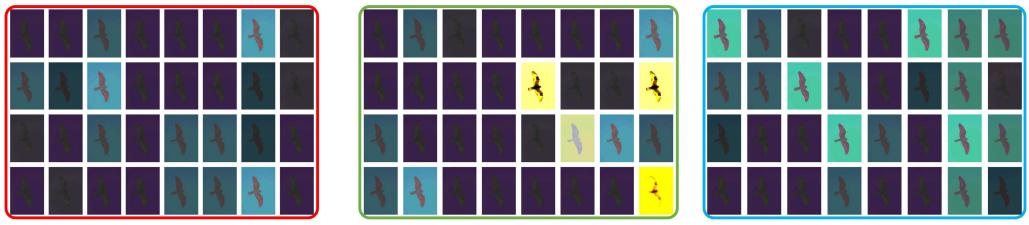

Figure 5. Comparison of SC, DSC, and GSConv output results.

Further, the convolutional kernel size of the DSC used in the original GSConv is 5×5, which is replaced with a 7×7 sized convolutional kernel to adapt it to the detection of large targets so we can obtain a larger scale of features and receptive field. This study reduces the network parameters and computation, minimizing the drawbacks of DSC, reducing its detrimental effects on the model, and making full use of the effective computational capacity of DSC to make the model easier to deploy to the endpoints.

3.4. VoVGSCSP

Based on a new hybrid convolutional approach, GSConv, we introduce a GS bottleneck based on Bottleneck and replace Bottleneck in C3 with GS bottleneck to improve C3. Bottleneck originally comes from Resnet and is proposed for high-level Resnet networks. It consists of three SCs with convolutional kernels of sizes 1×1, 3×3, and 1×1, respectively, where the 1×1 convolutional kernel serves to reduce and recover dimensionality, and the 3×3 is the bottleneck layer with smaller input and output dimensions. The special structure of Bottleneck means that it is easy to change dimensionality and achieve feature dimensionality reduction, thus, reducing the computational effort [46].

A comparison of the structure of Bottleneck and GS bottleneck is shown in Figure 6. Compared to Bottleneck, GS bottleneck replaces the two 1×1 SCs with GSConv and adds a new skip connection. The two branches of GS bottleneck, thus, perform separate convolutions without sharing weights and by splitting the number of channels so that the

number of channels is propagated via different network paths. The propagated channel information thus gains greater correlation and discrepancy, which not only ensures the accuracy of the information but also reduces the computational effort [47].

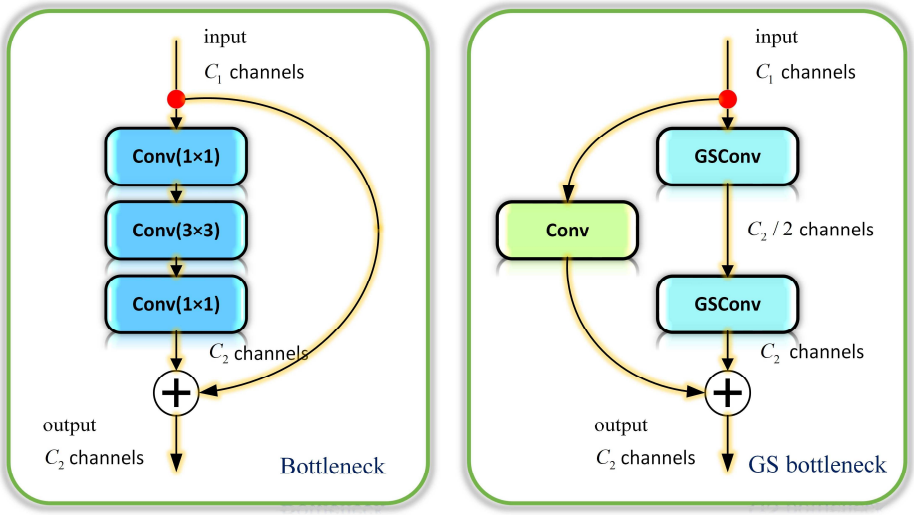

Figure 6. Structure of Bottleneck and GS bottleneck.

In this paper, we use an aggregation method to embed the GS bottleneck in C3 to replace Bottleneck and design a newly structured VoVGSCSP module. A comparison of the structure of C3 and VoVGSCSP is shown in Figure 7.

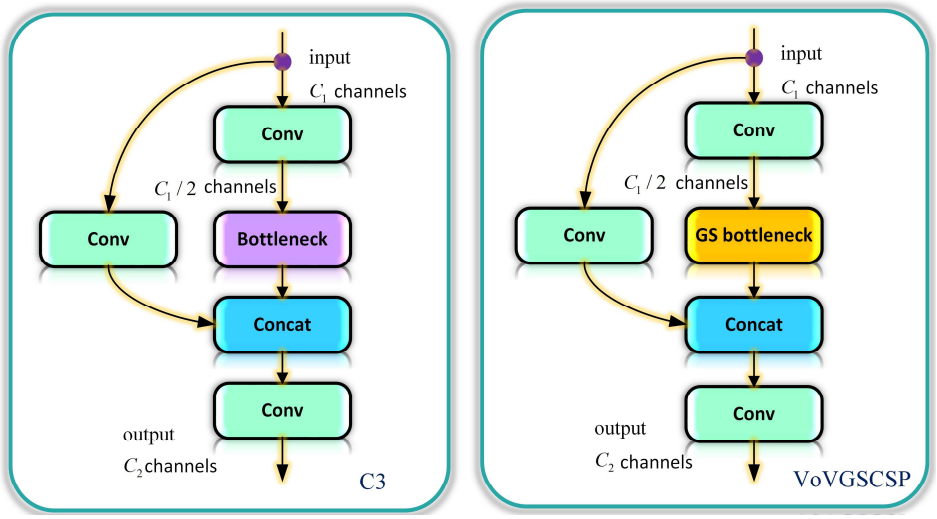

Figure 7. Structure of C3 and VoVGSCSP.

In VoVGSCSP, the input feature map splits the number of channels into two parts, the first part first passing through the Conv for convolution, after which the features are extracted by the stacked GS bottleneck module. The other part is connected as residuals and passes through only one Conv to convolve. The two parts are fused and connected according to the number of channels and finally output by Conv convolution. VoVGSCSP is not only compatible with all the advantages of GSConv but also has all the advantages that GS bottleneck brings. Thanks to the new skip-connected branch, VoVGSCSP has a stronger non-linear representation compared to C3, solving the problem of gradient disappearance. Meanwhile, similar to the segmentation gradient flow strategy of a cross-stage partial network (CSPNet), VoVGSCSP's split-channel approach enables rich gradient combinations, avoiding the repetition of gradient information and improving learning

ability. Ablation experimental results showed that VoVGSCSP reduces the computational effort and improves the accuracy of the model [48,49].

Combining the above improvements in the Backbone of YOLOv5s, we replace the SPP module with the SPPF module to improve the pooling efficiency while adding the Conv module to achieve 64× downsampling output; in the Head of YOLOv5s, we replace all the Conv modules with GSConv modules to reduce the number of parameters and computation brought by the upgrade of the feature pyramid structure. The C3 module is replaced with VoVGSCSP module, and the features are extracted by the stacked GS bottleneck for better compatibility with the GSConv module; at the same time, the original 8-fold downsampling feature fusion layer is deleted, and a 64-fold downsampling feature fusion layer is added to strengthen the learning capability of the network and give full play to the efficient computational capability of GSConv. The rest of the original modules of YOLOv5s remain unchanged [25]. Since all the improvements in this paper have good compatibility for different numbers of residual components and convolutional kernels and are not affected by the deepening of the network structure, for the YOLOv5m, YOLOv5l, and YOLOv5x models, which differ only at the network size and depth levels, the same improvements are also applicable. In this paper, we only take YOLOv5s as an example, and the improved network structure is shown in Figure 8.

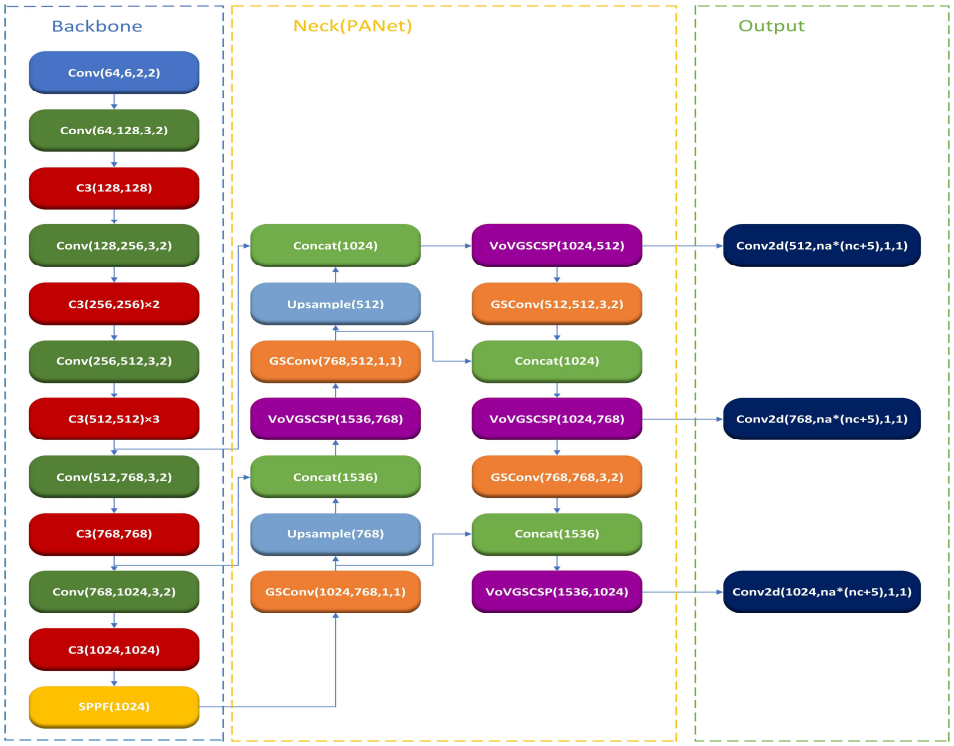

Figure 8. The network structure of the improved model in this paper.

4. Experiments and Analysis of Results

4.1. Experimental Environment

The software environment for the experiments is Linux Ubuntu 20.04, Pytorch 12.0 as the deep learning framework, CUDA 11.6, and Python 3.8.

The hardware environment for the experiments is Intel(R) Core (TM) i7-10750H CPU@2.60 GHz (12 CPUs), ~2.6 GHz, Nvidia Tesla A40, 48 G (From NVIDIA, Santa Clara, CA, USA).

4.2. Target Detection Experiments Based on Wildlife Datasets

Experimental Dataset

In the training phase, the image size is redefined in this paper as 640 × 640 to reduce the computational effort of a single image. The images are randomly cropped, randomly scaled, and randomly lined up. The dataset is enhanced and enriched by using the Mosaic data. The experimental weight decay is set to 0.0005, the learning rate to 0.015, and the number of iterations to 600. The wildlife dataset used for the experiments is sourced from the OpenImages public dataset, which covers real wildlife images in several scenarios. The dataset is annotated using labeling in XML format. There are a total of 2800 sample images in the dataset, so 1680 images are divided into the training set, 560 images into the test set, and 560 images into the validation set in a ratio of 6:2:2. The distribution of the constructed dataset is shown in Table 1.

Table 1. Distribution of datasets.

Experimental Datasets	Number
Training set	1680
Test set	560
Validation set	560
Total	2800

Figure 9 shows an example image of the wildlife dataset in this paper.

Figure 9. Example of wildlife dataset.

In this paper, precision, recall, AP (average precision), mAP (mean average precision), model parameters (Parameters), model operation (GFLOPs), and frames per second (FPS) are used to evaluate the model [50].

TP refers to the number of detected frames where IoU is greater than the set threshold (denoted as I, 0.5 in this paper, and the same true frame is only recorded for the first time) [51], while frames with IoU $\leq$ I are FP, i.e., the number of extra detected frames where the same true frame is detected, FN refers to the number of true frames that are not identified, and TN refers to the number of samples that are themselves negative and are also identified as negative ones. The confusion matrix is shown in Table 2.

Table 2. Confusion matrix relationship table.

Confusion Matrix		Predicted Condition	
		Positive Sample	**Negative Sample**
True condition	Positive sample	TP (True Positive)	FN (False Negative)
	Negative sample	FP (False Positive)	TN (True Negative)

Because of the possible limitations of precision and recall, the two need to be evaluated in combination, and the AP and mAP are evaluated while precision P and recall R are considered [52]. The formulae for each of these metrics are as follows:

$$precision = \frac{TP}{TP + FP} \tag{8}$$

$$recall = \frac{TP}{TP + FN} \tag{9}$$

Considering the possible limitations of precision and recall, neither metric is sufficient to evaluate model performance alone. Since the extent to which the model is affected by precision and recall respectively is unknown, to explore and measure both, we introduce a P–R curve, where the P in the P–R curve refers to precision, and R represents recall. The P–R curve represents the correlation between the precision rate and recall rate. In general, recall is set as the abscissa, and precision is set as the ordinate. AP is defined as the mean value of the precision rate for different recall rates, which is a measure that visually reflects the degree of model misidentification. It is calculated by finding the area under the P–R curve with the following formula:

$$AP = \int_0^1 P(R)dR \tag{10}$$

The mean average precision mAP represents the average accuracy of all species. The formula is as follows:

$$mAP = \frac{1}{N}\sum_{i=1}^{N} AP_i \tag{11}$$

mAP includes mAP@0.5 and mAP@0.5:0.95. mAP@0.5 is calculated when the IoU threshold value is 0.5. For one of the categories with n positive example samples, its mAP@0.5 is the average resulting value of the AP for these n samples. Increasing the IoU threshold from 0.5 in steps of 0.05 to 0.95 and taking the mean value of AP, their corresponding mAP@0.5 can show the trends in AP and R. The higher the mAP@0.5, the easier it is to maintain a high level of both AP and R. mAP@0.5:0.95 is the overall performance under different IoU thresholds, which takes the overall situation into account. A higher mAP@0.5:0.95 means that the model is more capable of high-precision boundary regression, i.e., the more accurate the fit of the prediction frame to the anchor frame. FPS represents the number of images that is detected per second. Supposing it takes t seconds to process each picture, the calculation formula is as follows:

$$FPS = \frac{1}{t} \tag{12}$$

4.3. Experimental Results and Analysis

The selection of the initial anchor frame in YOLOv5s is extremely important. After calculating the distance between the prediction frame and the real frame based on the initial anchor frame, the network must repeat in order to update the network parameters in the opposite direction. Adaptive anchor frame calculation can calculate the optimal anchor frame coordinates in the training set by an adaptive, iterative update with each instance of training. However, the outcome of such calculations is occasionally not ideal, so this paper uses the

genetic clustering method to conduct dimensional clustering analysis on the width and height of the target frame [53] to calculate new anchor values, which can speed up the model's convergence and boost recognition accuracy. The anchor boxes obtained by clustering are matched according to the feature map scale, and the results are shown in Table 3.

Table 3. Anchor box matching.

Feature Map	40 × 40	20 × 20	10 × 10
Anchors	(57,61) (107,113) (129,244)	(221,170) (231,322) (409,247)	(376,375) (368,534) (535,419)

Based on the object classification of the dataset, the sample images can be classified into five categories: antelope, elephant, leopard, eagle, and giraffe. A total of 560 images of each animal are selected, and we use the original YOLOv5s model and the improved YOLOv5s model to perform the experiments. The improved YOLOv5s target detection method proposed in this paper contains improvements to the feature pyramid network structure, anchors, and GSConv. To demonstrate the effectiveness of these three components, ablation experiments are performed on the animal dataset in this paper for these three improved components. To ensure the fairness of the experiments, the input image size is always kept at 640 × 640, and the hyperparameters are all set to be constant, and the experimental results are shown in Table 4.

Table 4. Ablation experiment.

Models	Size	Parameters/10^6	GFLOPs/10^9	mAP@0.5	mAP@0.5:0.95	Latency (ms)
YOLOv5s	640	7.02	15.8	82.2	52.9	2.1
+SPPF	640	7.02	15.8	82.4	54.5	1.5
Improvement	-	-	-	+0.2%	+1.6%	−28.6%
+P4–6	640	11.7	13.4	83.5	57.4	1.7
Improvement	-	+66.7%	−15.2%	+1.1%	+2.9%	+13.3%
+GSConv	640	10.6	13.0	83.4	57.6	1.4
Improvement	-	−9.40%	−3.0%	−0.1%	+0.2%	−11.8%
+VoVGSCSP	640	11.0	12.2	85.4	59.7	1.7
Improvement	-	+3.77%	−6.15%	+2%	+2.1%	+21.4%

(1) The effectiveness of SPPF. In this paper, the SPPF is improved in the first set of experiments based on the SPP by halving the number of channels of the feature map through the CBS, downsampling the maximum pooling layer, and replacing the original parallel maximum pooling layer with a series one, which improves the receptive field. Compared to SPP, SPPF has a higher detection speed with the same output results, for which a comparison is made below in this paper. As can be seen from Table 4, after improving SPP, mAP@0.5 increases by 0.2%, mAP@0.5:0.95 increases by 1.6%, and latency decreases by 28.6%. This is because SPPF solves the problem of multi-scale target fusion, and the receptive field is improved, which is conducive to improving target detection accuracy and speed;

(2) The effectiveness of P4–6. The feature pyramid network structure is stacked up one level overall in the second set of experiments in this paper to increase the receptive field to improve target detection performance. The new P6 target detection layer provides more sufficient large-size target information in the fusion process of feature information transfer to improve the feature fusion and feature extraction capability of the network. As can be seen from Table 4, after improving P4–6, mAP@0.5 increases by 1.1%, and mAP@0.5:0.95 increases by 2.9%, which is due to the higher downsampling

magnification of P6 and the larger receptive field per pixel point, and a larger receptive field can improve the target detection accuracy;

(3) Effectiveness of GSConv. This paper introduces the GSConv hybrid convolution module to replace the standard convolution in the Neck in the third set of experiments in order to decrease the number of parameters and computation of the model while retaining more channel information and enhancing the feature extraction and fusion capability of the network. GSConv stacks SC and DSC on top of the lightweight network so that SC and DSC are feature connected. Additionally, it makes use of ShuffleNets to enable the fusion of channel information from SC and feature information from DSC into the output of DSC for channel information interaction. This lessens the negative effects of DSC on the model while maintaining a lower number of model parameters, reducing computational effort, and minimizing the loss of channel information. As can be seen from Table 4, the introduction of GSConv results in an all-round improvement in the detection performance of the model, with mAP@0.5 and mAP@0.5:0.95 remaining largely unchanged, while Parameters are reduced by 9.4%, GFLOPs by 3.0%, and latency by 11.8%;

(4) Effectiveness of VoVGSCSP. In this paper, we use VoVGSCSP to replace C3 in the fourth set of experiments and design the GS bottleneck module based on GSConv, splitting the number of channels so that information is passed through different paths, reducing the computational effort of the original Bottleneck module. VOVGSCSP replaces Bottleneck with the GS bottleneck and embeds it in C3. The GS bottleneck splits the number of channels and adds a new branch through Conv convolution, and the two parts are then feature connected. This method of splitting the number of channels enables a rich combination of gradients, avoiding repetition of gradient information and improving learning ability. It can also enhance the non-linear representation of the model and improve its accuracy. As can be seen from Table 4, the detection accuracy of the model is improved by using VoVGSCSP; mAP@0.5 increases by 2%, and mAP@0.5:0.95 increases by 2.1%, while the computational effort of the model is also reduced, with a 6.15% reduction in GFLOPs.

In addition, in order to prove the superiority of the experimental results, we synthesize multiple sets of experimental results and compare them together. Figure 10 shows the visualization of the ablation experimental data of the improved methods.

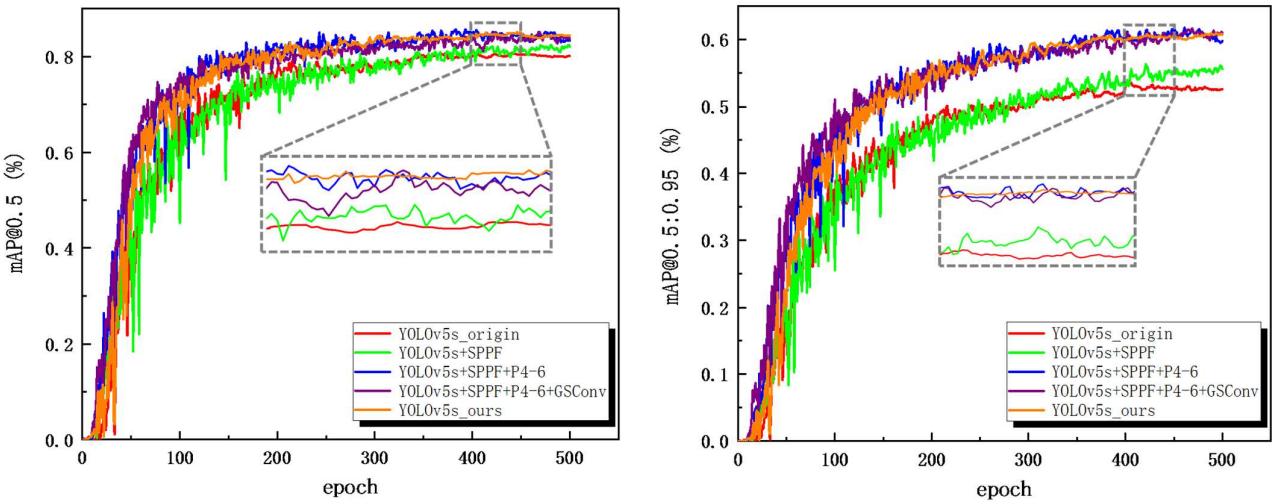

Figure 10. The ablation experimental results of the improved methods.

The AP values of each model in the ablation experiments for individual categories are shown in Table 5. The experimental results show that the detection accuracy is higher when the wild animals themselves are clearly distinguished from their environment. The improved YOLOv5s detection algorithm in this paper gives higher performance for certain animals that

are large targets, such as elephants and giraffes, and the AP values of elephants are improved by 5.6% and giraffes by 5.2%. The images of eagles in the dataset of this paper are mostly small targets, while, in the model of this paper, the pyramidal network structure is stacked one level up overall, which improves the perceptual field and enhances the detection ability for large targets, and the detection ability for small targets may be weaker, but it still maintains high accuracy. Compared with other classical algorithms in the YOLO series, such as YOLOv3-tiny and YOLOv4-tiny, the improved YOLOv5s detection algorithm in this paper has advantages for the detection accuracy of all five types of animals, and, overall, the improved YOLOv5s detection algorithm in this paper has a performance that surpasses other algorithms in the same series and can better perform the target detection task.

Table 5. The AP value of each model in a single category in the ablation experiment.

Method	Antelope	Eagle	Elephant	Giraffe	Leopard
YOLOv3-tiny	0.574	0.783	0.609	0.687	0.630
YOLOv4-tiny	0.521	0.784	0.566	0.645	0.614
YOLOv6s	0.731	0.809	0.766	0.818	0.936
YOLOv7-tiny	0.739	0.825	0.720	0.878	0.869
YOLOv5s	0.723	0.911	0.781	0.814	0.892
YOLOv5s-SF	0.772	0.805	0.780	0.858	0.901
YOLOv5s-SF + P4–6	0.764	0.823	0.845	0.880	0.911
YOLOv5s-SF + P4–6 + GSConv	0.799	0.854	0.868	0.857	0.879
YOLOv5s-SF + P4–6 + GSConv + VoVGSCSP	0.765	0.867	0.837	0.866	0.903

The improved YOLOv5s algorithm has better robustness and environmental adaptability, and the detection accuracy is further improved. Since the five animals analyzed in this paper are similar to most animals in nature, it is, therefore, feasible to extend the model to other animals for classification, and the model in this paper can be better applied to the detection of wild animals. This also verifies that the improvement of the network structure of YOLOv5s in this paper makes the model improve the mAP of all the sample images.

4.4. Comparative Analysis of Algorithms

4.4.1. Comparison with Mainstream Target Detection Algorithms

To objectively evaluate the performance of the model, we conduct a side-by-side comparison with the mainstream target detection models YOLOv3-tiny, YOLOv4-tiny, Faster R-CNN, and SSD through the YOLOv5s before improvement as well as through the YOLOv5s after improvement. We also use mAP@0.5 and FPS for comparison, which further verify that the improved YOLOv5s algorithm in this paper outperforms other target detection algorithms in detecting wild animals in natural environments.

The datasets in this paper are applied to the YOLOv3-tiny, YOLOv4-tiny, Faster R-CNN + VGG16, SSD + VGG16, YOLOv6s, and YOLOv7-tiny target detection algorithms for experiments [54,55], all of which are performed independently. We use mAP@0.5, FPS, and GFLOPs [56,57] for comparison; the results are shown in Table 6. By comparing the final analysis, it can be obtained that, for both the detection accuracy and the detection speed of the models, on the animal dataset used in this paper, the mAP@0.5 and FPS of the improved YOLOv5s are superior to other mainstream target detection algorithms. Furthermore, the GFLOPs of the improved model reaches the lowest of all the mainstream target detection models.

4.4.2. Comparison of the Detection Effect of the Model before and after the Improvement

Six sample images are taken from the test set of the improved YOLOv5s model before and after the comparison, and the results are shown in Figure 11.

Table 6. Object detection model performance comparison.

Model	mAP@0.5	FPS	GFLOPs
YOLOv3-tiny	65.7	87.9	12.9
YOLOv4-tiny	62.6	99.6	20.6
Faster R-CNN + VGG16	74.0	40.1	172
SSD + VGG16	82.5	65.4	31
YOLOv6s	80.1	109.3	44.07
YOLOv7-tiny	81.7	107.2	13.1
YOLOv5s	82.2	102.5	15.8
YOLOv5s-SF	82.4	109.4	15.8
YOLOv5s-SF + P4–6	83.5	107.3	13.4
YOLOv5s-SF + P4–6 + GSConv	83.4	119.8	13.0
YOLOv5s-SF + P4–6+ GSConv + VoVGSCSP	85.4	111.9	12.2

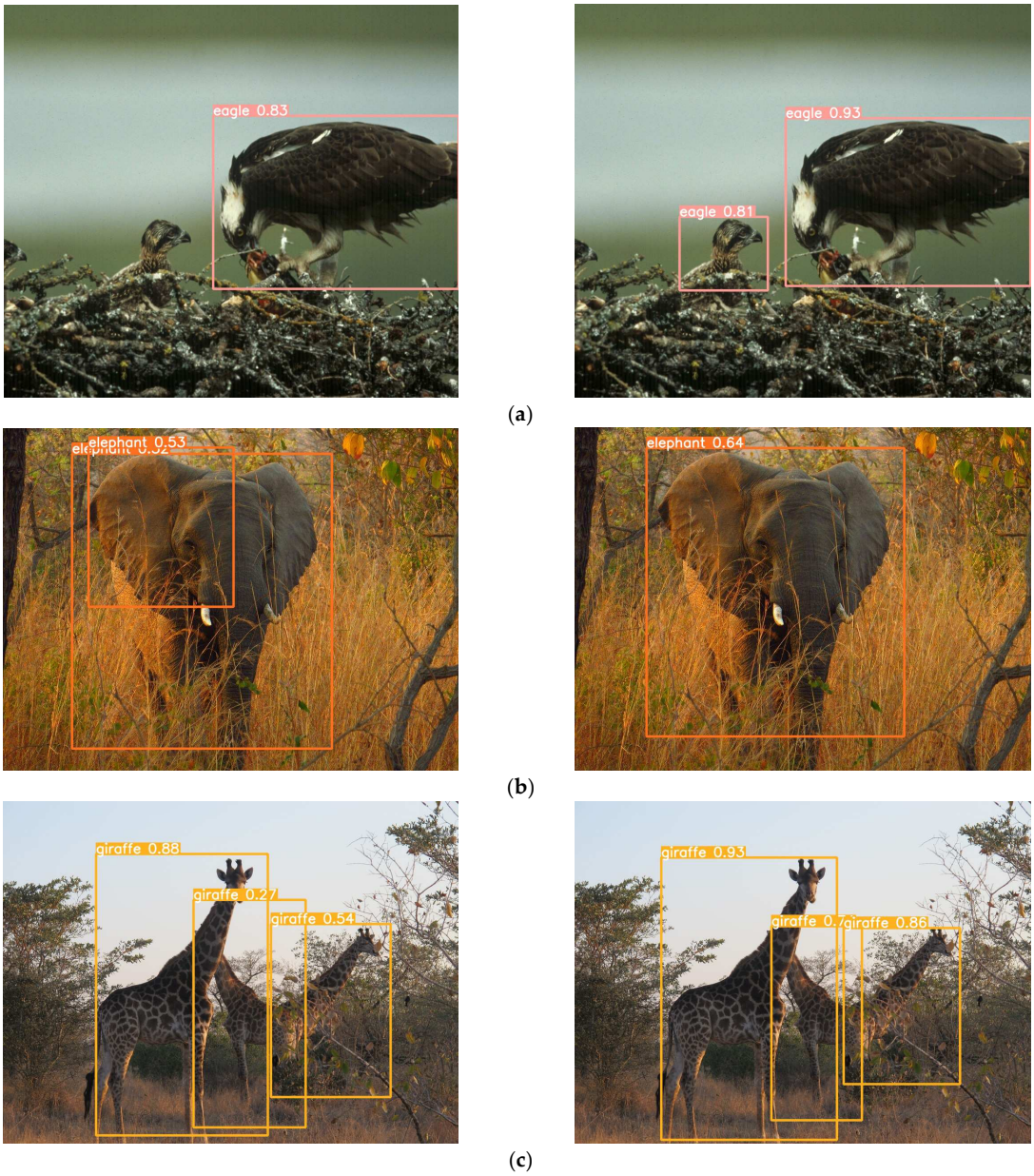

Figure 11. Example of comparison of detection results before and after improvement. (**a**) Detection effect of eagles in the model before and after the improvement; (**b**) Detection effect of elephant in the model before and after the improvement; (**c**) Detection effect of giraffes in the model before and after the improvement.

From the comparison diagram in Figure 11a, it can be seen that the original YOLOv5s model does not detect the eagle on the left, while the improved model does not miss the eagle, and the detection accuracy of the original model is further improved. From the comparison diagram in Figure 11b, it can be seen that the elephant is a large target detected in this figure. Since the newly added P6 target detection layer provides more sufficient large-size target information during the fusion process of feature information transfer, increasing the receptive field, the model before the improvement recognizes one elephant as two, and the model after the improvement has no false detection, which reflects the further improvement of the large target detection ability. From the comparison chart in Figure 11c, it can be seen that, in the case of three giraffes with body overlap, the detection accuracy of the improved model for giraffes is higher than that of the original model. The upgraded YOLOv5s model outperforms the original model in detection and recognition, and it lowers the rate of missed detection and false detection of target animals according to the study of the sample picture detection findings.

4.4.3. Comparison of the Detection Effect of the Model before and after Improvement on the VOC2007 + 2012 Dataset

In order to test the generalization ability of the model, we further test the detection ability of the improved model in this paper on other public datasets so as to judge the recognition ability of the improved model for targets of various sizes. In this paper, the Pascal VOC2007 + 2012 public dataset [58] is selected for testing experiments, in which the proportion of small targets is increased to better test the model's ability to detect small targets. A comparison of the performance of the original YOLOv5s and the improved model in this paper on the Pascal VOC2007 + 2012 public dataset is shown in Table 7.

Table 7. Performance comparison between YOLOv5s and YOLOv5_ours on Pascal VOC2007 + 2012 datasets.

Models	Size	Parameters/10^6	GFLOPs/10^9	mAP@0.5	mAP@0.5:0.95	Latency (ms)
YOLOv5s	640	7.06	16.0	63.6	41.6	1.5
YOLOv5_ours	640	11.1	12.3	69.1	49.2	1.4
Improvement	-	+57.22%	−23.13%	+5.5%	+7.6%	−6.67%

As can be seen from Table 7, the improved YOLOv5s model in this paper improves the detection accuracy and comprehensive performance in the Pascal VOC2007 + VOC2012 public dataset, and the improved method is still applicable to the detection of small target objects. Other indicators of this experiment and simulation results are shown in Figures 12 and 13.

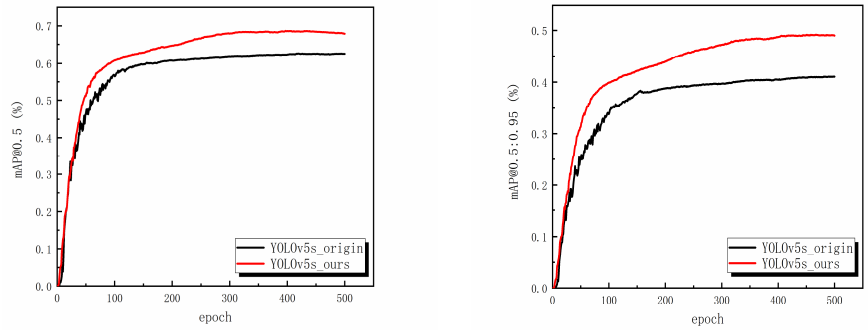

Figure 12. The experimental results of the improved methods on Pascal VOC2007 + VOC2012 datasets.

Figure 13. Comparison of simulation results of YOLOv5s and YOLOv5s_ours on Pascal VOC2007 + VOC2012 dataset.

5. Conclusions

In this paper, YOLOv5s is better applied to large datasets of animals in natural environments by improving the feature pyramid network structure and convolution module, which effectively improves the detection accuracy and speed of YOLOv5s model on this animal dataset. The final experimental findings demonstrate that the updated YOLOv5s algorithm's detection accuracy and model performance are enhanced to varying degrees when compared to other mainstream networks in the same experimental setting. The final improved YOLOv5s algorithm has a mAP@0.5 of 85.4% for the animal dataset in this paper and a mAP@0.5:0.95 of 59.7%. Compared with the original YOLOv5s, the improved model mAP@0.5 increases by 3.2%, mAP@0.5:0.95 increases by 6.8%, and GFLOPs decreases by 22.78%. While increasing the number of model parameters, FPS increases by 9.4. It can be seen that the model in this paper is improved in both detection accuracy and model lightness for large animal recognition. The accuracy and real-time performance of the detection meet the demand and can finally achieve high-accuracy real-time detection with a small amount of model calculation. In actual natural environment tests, our network structure needs to further improve the detection ability of some wild animals due to their mimetic ability, and, considering the simultaneous occurrence of multiple wild animals, further spatial features need to be extracted to obtain the correlation between different targets in space. The next phase of study will, therefore, concentrate on further network structure optimization and the addition of an attention mechanism to enhance performance in this area.

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

Funding: This research was funded by National Innovation and Entrepreneurship Training Program for College Students, China, grant number S202210497212, S202210497068 and the National Natural Science Foundation of China, grant number 91324201.

Data Availability Statement: Publicly available datasets were analyzed in this study. This data can be found here: [https://gitee.com/that-wipe-of-light/wildlife-object-detection-method-applying-segmentation-gradient-flow-and-feature-dimensionality-reduction (accessed on 25 November 2022)].

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

References

1. Han, Y.; Chen, L.; Luo, Y.; Ai, H.; Hong, Z.; Ma, Z.; Wang, J.; Zhou, R.; Zhang, Y. Underwater Holothurian Target-Detection Algorithm Based on Improved CenterNet and Scene Feature Fusion. *Sensors* **2022**, *22*, 7204. [CrossRef] [PubMed]
2. Luo, X.; Wang, Y.; Cai, B.; Li, Z. Moving Object Detection in Traffic Surveillance Video: New MOD-AT Method Based on Adaptive Threshold. *ISPRS Int. J. Geo-Inf.* **2021**, *10*, 742. [CrossRef]
3. Xu, X.; Zhao, M.; Shi, P.; Ren, R.; He, X.; Wei, X.; Yang, H. Crack Detection and Comparison Study Based on Faster R-CNN and Mask R-CNN. *Sensors* **2022**, *22*, 1215. [CrossRef] [PubMed]
4. Iftikhar, S.; Zhang, Z.; Asim, M.; Muthanna, A.; Koucheryavy, A.; Abd El-Latif, A.A. Deep Learning-Based Pedestrian Detection in Autonomous Vehicles: Substantial Issues and Challenges. *Electronics* **2022**, *11*, 3551. [CrossRef]
5. Akhtar, M.J.; Mahum, R.; Butt, F.S.; Amin, R.; El-Sherbeeny, A.M.; Lee, S.M.; Shaikh, S. A Robust Framework for Object Detection in a Traffic Surveillance System. *Electronics* **2022**, *11*, 3425. [CrossRef]
6. Cong, P.; Lv, K.; Feng, H.; Zhou, J. Improved YOLOv3 Model for Workpiece Stud Leakage Detection. *Electronics* **2022**, *11*, 3430. [CrossRef]
7. Jiang, S.; Zhou, X. DWSC-YOLO: A Lightweight Ship Detector of SAR Images Based on Deep Learning. *J. Mar. Sci. Eng.* **2022**, *10*, 1699. [CrossRef]
8. Jiang, X.; Sun, K.; Ma, L.; Qu, Z.; Ren, C. Vehicle Logo Detection Method Based on Improved YOLOv4. *Electronics* **2022**, *11*, 3400. [CrossRef]
9. Mallela, N.C.; Volety, R.; Perumal, S.R.; Nadesh, R.K. Detection of the triple riding and speed violation on two-wheelers using deep learning algorithms. *Multimed. Tools Appl.* **2020**, *80*, 8175–8187. [CrossRef]
10. Rim, B.; Kim, J.; Hong, M. Fingerprint classification using deep learning approach. *Multimed. Tools Appl.* **2020**, *80*, 35809–35825. [CrossRef]
11. Yao, J.; Fan, X.; Li, B.; Qin, W. Adverse Weather Target Detection Algorithm Based on Adaptive Color Levels and Improved YOLOv5. *Sensors* **2022**, *22*, 8577. [CrossRef]
12. Wang, H.; Sun, S.; Wu, X.; Li, L.; Zhang, H.; Li, M.; Ren, P. A YOLOv5 baseline for underwater object detection. In Proceedings of the OCEANS 2021: San Diego—Porto, San Diego, CA, USA, 20–23 September 2021; pp. 1–4.
13. Sun, S.; Wang, H.; Zhang, H.; Li, M.; Xiang, M.; Luo, C.; Ren, P. Underwater Image Enhancement with Reinforcement Learning. *IEEE J. Ocean. Eng.* **2022**, 1–13. [CrossRef]
14. Liu, W.; Xiao, Y.; Zheng, A.; Zheng, Z.; Liu, X.; Zhang, Z.; Li, C. Research on Fault Diagnosis of Steel Surface Based on Improved YOLOV5. *Processes* **2022**, *10*, 2274. [CrossRef]
15. Wang, F.; Sun, Z.; Chen, Y.; Zheng, H.; Jiang, J. Xiaomila Green Pepper Target Detection Method under Complex Environment Based on Improved YOLOv5s. *Agronomy* **2022**, *12*, 1477. [CrossRef]
16. Chandrakar, R.; Raja, R.; Miri, R. Animal detection based on deep convolutional neural networks with genetic segmentation. *Multimed. Tools Appl.* **2021**, *81*, 42149–42162. [CrossRef]
17. Meena, S.D.; Agilandeeswari, L. Smart Animal Detection and Counting Framework for Monitoring Livestock in an Autonomous Unmanned Ground Vehicle Using Restricted Supervised Learning and Image Fusion. *Neural Process. Lett.* **2021**, *53*, 1253–1285. [CrossRef]
18. Chen, J.; Wu, Y.; Yang, Y.; Wen, S.; Shi, K.; Bermak, A.; Huang, T. An Efficient Memristor-Based Circuit Implementation of Squeeze-and-Excitation Fully Convolutional Neural Networks. *IEEE Trans. Neural Netw. Learn. Syst.* **2022**, *33*, 1779–1790. [CrossRef]
19. Lyu, B.; Hamdi, M.; Yang, Y.; Cao, Y.; Yan, Z.; Li, K.; Wen, S.; Huang, T. Efficient Spectral Graph Convolutional Network Deployment on Memristive Crossbars. *IEEE Trans. Emerg. Top. Comput. Intell.* **2022**, 1–11. [CrossRef]
20. Lee, S.; Moon, W.; Heo, J.-P. Task discrepancy maximization for fine-grained few-shot classification. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022; pp. 5321–5330.
21. Li, H.; Cui, Z.; Liu, S.; Tan, P.; Fraser, S. RAGO: Recurrent graph optimizer for multiple rotation averaging. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022; pp. 15766–15775.
22. Yeh, Y.-Y.; Li, Z.; Hold-Geoffroy, Y.; Zhu, R.; Xu, Z.; Hašan, M.; Sunkavalli, K.; Chandraker, M. PhotoScene: Photorealistic material and lighting transfer for indoor scenes. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022; pp. 18541–18550.
23. Liau, Y.Y.; Ryu, K. Status Recognition Using Pre-Trained YOLOv5 for Sustainable Human-Robot Collaboration (HRC) System in Mold Assembly. *Sustainability* **2021**, *13*, 12044. [CrossRef]
24. Walia, I.S.; Kumar, D.; Sharma, K.; Hemanth, J.D.; Popescu, D.E. An Integrated Approach for Monitoring Social Distancing and Face Mask Detection Using Stacked ResNet-50 and YOLOv5. *Electronics* **2021**, *10*, 2996. [CrossRef]

25. Lamane, M.; Tabaa, M.; Klilou, A. Classification of targets detected by mmWave radar using YOLOv5. *Procedia Comput. Sci.* **2022**, *203*, 426–431. [CrossRef]
26. Zhao, Z.; Yang, X.; Zhou, Y.; Sun, Q.; Ge, Z.; Liu, D. Real-time detection of particleboard surface defects based on improved YOLOV5 target detection. *Sci. Rep.* **2021**, *11*, 21777. [CrossRef] [PubMed]
27. Chang, Y.H.; Zhang, Y.Y. Deep Learning for Clothing Style Recognition Using YOLOv5. *Micromachines* **2022**, *13*, 1678. [CrossRef] [PubMed]
28. Peng, L.; Li, B.; Yu, W.-H.; Yang, K.; Shao, W.; Wang, H. SOTIF Entropy: Online SOTIF Risk Quantification and Mitigation for Autonomous Driving. *arXiv* **2022**, arXiv:2211.04009.
29. Hao, Y.; Pei, H.; Lyu, Y.; Yuan, Z.; Rizzo, J.-R.; Wang, Y.; Fang, Y. Understanding the Impact of Image Quality and Distance of Objects to Object Detection Performance. *arXiv* **2022**, arXiv:2209.08237.
30. Haque, M.E.; Rahman, A.; Junaeid, I.; Hoque, S.U.; Paul, M. Rice Leaf Disease Classification and Detection Using YOLOv5. *arXiv* **2022**, arXiv:2209.0157.
31. Wu, Y.; Sun, Y.; Zhang, S.; Liu, X.; Zhou, K.; Hou, J. A Size-Grading Method of Antler Mushrooms Using YOLOv5 and PSPNet. *Agronomy* **2022**, *12*, 2601. [CrossRef]
32. Li, H.; Yang, G. Dietary Nutritional Information Autonomous Perception Method Based on Machine Vision in Smart Homes. *Entropy* **2022**, *24*, 868. [CrossRef]
33. Zhu, Y.; Yan, W.Q. Traffic sign recognition based on deep learning. *Multimed. Tools Appl.* **2022**, *81*, 17779–17791. [CrossRef]
34. Chen, S.; Duan, J.; Wang, H.; Wang, R.; Li, J.; Qi, M.; Duan, Y.; Qi, S. Automatic detection of stroke lesion from diffusion-weighted imaging via the improved YOLOv5. *Comput. Biol. Med.* **2022**, *150*, 106120. [CrossRef]
35. Majeed, F.; Khan, F.Z.; Nazir, M.; Iqbal, Z.; Alhaisoni, M.; Tariq, U.; Khan, M.A.; Kadry, S. Investigating the efficiency of deep learning based security system in a real-time environment using YOLOv5. *Sustain. Energy Technol. Assess.* **2022**, *53*, 102603. [CrossRef]
36. Du, S.; Zhang, B.; Zhang, P.; Xiang, P.; Xue, H. FA-YOLO: An Improved YOLO Model for Infrared Occlusion Object Detection under Confusing Background. *Wirel. Commun. Mob. Comput.* **2021**, *2021*, 1896029. [CrossRef]
37. Liu, H.; Sun, F.; Gu, J.; Deng, L. SF-YOLOv5: A Lightweight Small Object Detection Algorithm Based on Improved Feature Fusion Mode. *Sensors* **2022**, *22*, 5817. [CrossRef]
38. Xue, Z.; Lin, H.; Wang, F. A Small Target Forest Fire Detection Model Based on YOLOv5 Improvement. *Forests* **2022**, *13*, 1332. [CrossRef]
39. Bilecen, B.B.; Fisne, A.; Ayazoglu, M. Efficient multi-purpose cross-attention based image alignment block for edge devices. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), New Orleans, LA, USA, 19–20 June 2022; pp. 3638–3647.
40. Quan, Y.; Zhang, D.; Zhang, L.; Tang, J. Centralized Feature Pyramid for Object Detection. *arXiv* **2022**, arXiv:2210.02093.
41. Chen, Z.; Li, X.; Wang, L.; Shi, Y.; Sun, Z.; Sun, W. An Object Detection and Localization Method Based on Improved YOLOv5 for the Teleoperated Robot. *Appl. Sci.* **2022**, *12*, 11441. [CrossRef]
42. Rezaei, M.; Azarmi, M.; Mir, F.M.P. Traffic-Net: 3D Traffic Monitoring Using a Single Camera. *arXiv* **2022**, arXiv:2109.09165. [CrossRef]
43. Gupta, P.; Dixit, M. Image-based crack detection approaches: A comprehensive survey. *Multimed. Tools Appl.* **2022**, *81*, 40181–40229. [CrossRef]
44. Sun, X.; Hassani, A.; Wang, Z.; Huang, G.; Shi, H. DiSparse: Disentangled sparsification for multitask model compression. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022; pp. 12372–12382.
45. Mao, G.; Liao, G.; Zhu, H.; Sun, B. Multibranch Attention Mechanism Based on Channel and Spatial Attention Fusion. *Mathematics* **2022**, *10*, 4150. [CrossRef]
46. Tian, Z.; Shen, C.; Wang, X.; Chen, H. BoxInst: High-performance instance segmentation with box annotations. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 5439–5448.
47. Zhou, J.; Li, W.; Fang, H.; Zhang, Y.; Pan, F. The hull structure and defect detection based on improved YOLOv5 for mobile platform. In Proceedings of the 41st Chinese Control Conference (CCC), Hefei, China, 25–27 July 2022; pp. 6392–6397.
48. Yang, Z.; Li, L.; Luo, W. PDNet: Improved YOLOv5 Nondeformable Disease Detection Network for Asphalt Pavement. *Comput. Intell. Neurosci.* **2022**, *2022*, 5133543. [CrossRef]
49. Wu, F.; Duan, J.; Ai, P.; Chen, Z.; Yang, Z.; Zou, X. Rachis detection and three-dimensional localization of cut off point for vision-based banana robot. *Comput. Electron. Agric.* **2022**, *198*, 107079. [CrossRef]
50. Krosney, A.E.; Sotoodeh, P.; Henry, C.J.; Beck, M.A.; Bidinosti, C.P. Inside Out: Transforming Images of Lab-Grown Plants for Machine Learning Applications in Agriculture. *arXiv* **2022**, arXiv:2211.02972.
51. Raza, M.; Prokopova, H.; Huseynzade, S.; Azimi, S.; Lafond, S. SimuShips—A High Resolution Simulation Dataset for Ship Detection with Precise Annotations. *arXiv* **2022**, arXiv:2211.05237.
52. Dulal, R.; Zheng, L.; Kabir, M.A.; McGrath, S.R.; Medway, J.; Swain, D.L.; Swain, W. Automatic Cattle Identification using YOLOv5 and Mosaic Augmentation: A Comparative Analysis. *arXiv* **2022**, arXiv:2210.11939.
53. Rani, M.; Gagandeep. Effective network intrusion detection by addressing class imbalance with deep neural networks multimedia tools and applications. *Multimed. Tools Appl.* **2022**, *81*, 8499–8518. [CrossRef]

54. Li, C.; Li, L.; Jiang, H.; Weng, K.; Geng, Y.; Li, L.; Ke, Z.; Li, Q.; Cheng, M.; Nie, W.; et al. YOLOv6: A Single-Stage Object Detection Framework for Industrial Applications. *arXiv* **2022**, arXiv:2209.02976.
55. 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. *arXiv* **2022**, arXiv:2207.02696.
56. Lyu, B.; Wen, S.; Shi, K.; Huang, T. Multiobjective Reinforcement Learning-Based Neural Architecture Search for Efficient Portrait Parsing. *IEEE Trans. Cybern.* **2021**, 1–12. [CrossRef]
57. Lyu, B.; Yang, Y.; Wen, S.; Huang, T.; Li, K. Neural Architecture Search for Portrait Parsing. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, 1–10. [CrossRef]
58. Li, A. Slim-Neck by GSConv: A Better Design Paradigm of Detector Architectures for Autonomous Vehicles. Available online: https://github.com/AlanLi1997/slim-neck-by-gsconv/tree/master/datasets/VOC2012 (accessed on 2 January 2023).