applied sciences

MDPI

Article

Fast Rail Fastener Screw Detection for Vision-Based Fastener Screw Maintenance Robot Using Deep Learning

Yijie Cai [1,2], Ming He [1], Qi Tao [1,2], Junyong Xia [1,2], Fei Zhong [1,2] and Hongdi Zhou [1,2,*]

[1] School of Mechanical Engineering, Hubei University of Technology, Wuhan 430068, China; yijie.cai@hbut.edu.cn (Y.C.); 102210143@hbut.edu.cn (M.H.); taoqi@hbut.edu.cn (Q.T.); 20171013@hbut.edu.cn (J.X.); zhong_fei_hbut@163.com (F.Z.)

[2] Key Laboratory of Modern Manufacturing Quality Engineering in Hubei Province, Wuhan 430068, China

* Correspondence: zh_hongdi@163.com; Tel.: +86-027-59750418

Abstract: Fastener screws are critical components of rail fasteners. For the fastener screw maintenance robot, an image-based fast fastener screw detection method is urgently needed. In this paper, we propose a light-weight model named FSS-YOLO based on YOLOv5n for rail fastener screw detection. The C3Fast module is presented to replace the C3 module in the backbone and neck to reduce Params and FLOPs. Then, the SIoU loss is introduced to enhance the convergence speed and recognition accuracy. Finally, for the enhancement of the screw detail feature fusion, the shuffle attention (SA) is incorporated into the bottom-up process in the neck part. Experiment results concerning CIoU and DIoU for loss, MobileNetv3 and GhostNet for light-weight improvement, simple attention mechanism (SimAM), and squeeze-and-excitation (SE) attention for the attention module, and YOLO series methods for performance comparison are listed, demonstrating that the proposed FSS-YOLO significantly improves the performance, with higher accuracy and lower computation cost. It is demonstrated that the FSS-YOLO is 7.3% faster than the baseline model in FPS, 17.4% and 19.5% lower in Params and FLOPs, respectively, and the P, mAP@50, Recall, and F1 scores are increased by 10.6% and 6.4, 13.4%, and 12.2%, respectively.

Keywords: fastener screw maintenance robot; rail fastener screw detection; light weight; YOLO

Citation: Cai, Y.; He, M.; Tao, Q.; Xia, J.; Zhong, F.; Zhou, H. Fast Rail Fastener Screw Detection for Vision-Based Fastener Screw Maintenance Robot Using Deep Learning. *Appl. Sci.* **2024**, *14*, 3716. https://doi.org/10.3390/app14093716

Academic Editor: Alexandre Carvalho

Received: 18 March 2024
Revised: 12 April 2024
Accepted: 23 April 2024
Published: 26 April 2024

1. Introduction

As critical components of rail fasteners, the reliability of fastener screws is important for railways. The workers' eye is usually used for accurately positioning the internal combustion power screw wrench and the fastener screw, which is extremely low efficiency. Fastener screw maintenance robots that use computer vision technology can accurately identify and locate fastener screws, as shown in Figure 1, which can considerably reduce the work intensity and improve efficiency. The computing power of the equipment is restricted by the light weight, small size, and limited battery capacity of the robot, and it is challenging to keep high-performance devices working for a long time in various harsh environments. Therefore, a light-weight and high-performance fastener screw detection model is necessary to fulfill the detection task requirement.

The deep-learning method has been widely used for object detection applications [1–4]. Although two-stage approaches like R-CNN [5], Faster-RCNN [6,7], and Mask-RCNN [8,9] can achieve high detection accuracy, the computation cost is high, which impedes the usage in real-time fastener screw detection tasks. The one-stage methods, like the YOLO series [10–14], can obtain a fast detection speed with high accuracy.

To date, the application of a fast fastener screw detection method using deep learning in the railway maintenance field is rare. Mushtaq et al. [15] proposed a screw recognition method based on deep learning and image processing, using the YOLOv5 algorithm to classify and recognize bolts, nuts, etc. Zhuang et al. [16] proposed a two-stage target recognition method based on deep learning, further improving the detection performance.

The recognition task is on a railway and two kinds of fasteners. At the first stage, a modified YOLOv3 model is developed to provide the initial detection information. In the second stage, a domain logic-based hybrid model (DLHM) is introduced. The DLHM consists of the problem region calibration module and the symmetric region calibration module. He [17] proposed a nut localization and recognition method based on computer vision technology to accomplish automatic bolt assembly in automatic production lines. An industrial camera is used to supply the image of the nut, and then the image is processed using edge detection and Hough circle transformation to obtain the coordinates of the nut. Finally, the precise position of the target is obtained according to the nut's coordinates.

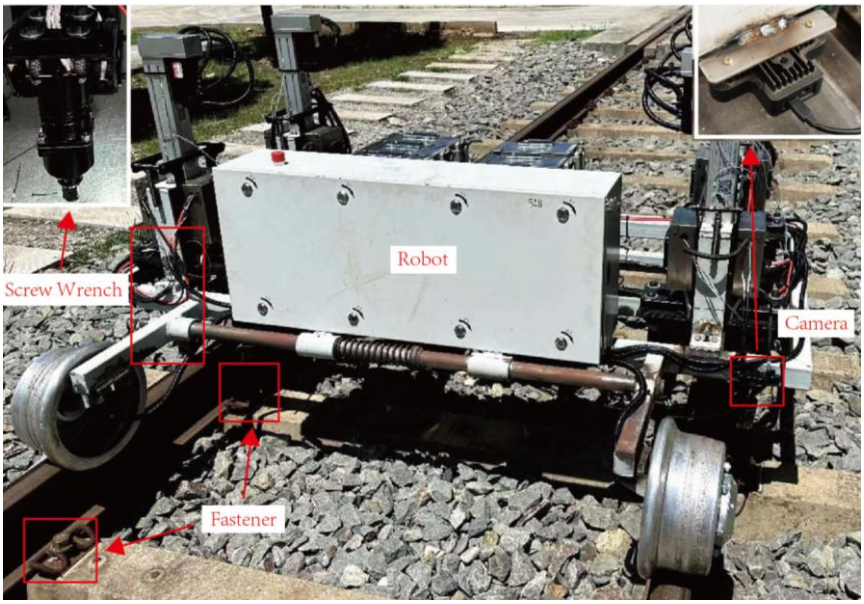

Figure 1. Fastener screw maintenance robot: The small picture in the upper left corner is the electric screw wrench, which is adapted for disassembling and assembling the fastener screws. The small picture in the upper right corner is the depth camera, which is used for the detection and location of fastener screws.

In the field of model performance enhancement, Panigrahi et al. [18] proposed a DSM-IDM-YOLO, which integrates depth-wise separable convolution modules (DSMs) and an inception depth-wise convolution module (IDM) to capture a wide range of features of the target. These modules are incorporated into different layers of YOLOv2 to enhance the model performance. Based on YOLOv3, Ma et al. [19] proposed an AVS-YOLO, which can improve the model performance by introducing a densely connected feature pyramid and a scale-aware attention module. Jiang et al. [20] proposed a light-weight real-time object detection model. In the backbone section, they introduced a receptive field-extended backbone with a restricted number of convolution layers to extract informative features. In the neck section, the feature pyramid network (FPN) incorporates additional lateral connections to recycle features within each convolutional stage, and both the channel attention and spatial attention modules are introduced, thus boosting the model's performance. Yin et al. [21] proposed a fast detection technique for the condition of elderly fall action based on an enhanced version of YOLOv5s. Asymmetric convolution blocks (ACBs) are utilized in the backbone network instead of the original Conv to enhance the feature extraction process. Then, they introduced a spatial attention mechanism module to improve the detection accuracy. Liu et al. [22] proposed a distracted driving detection model based on YOLOv7. The modified model can boost the overall performance of YOLOv7 by adopting the global attention mechanism and channel-based data augmentation. Yang et al. [23] proposed an effective steel strip surface defect detection model. Its feature fusion process is replaced with the de-weighted BiFPN structure, maximizes the utilization of feature

information, and minimizes the loss of valuable data. Subsequently, the integration of the ECA attention mechanism reinforces crucial feature channels. Finally, the SIoU is adopted as a substitute for the conventional IoU to elevate the model's performance.

Though the research regarding accuracy enhancement methods has obtained many achievements, there remains vast space for improvement of the computation speed. In this paper, a light-weight fastener screw detection model named FSS-YOLO is proposed. Compared with other competitive models, our work demonstrates a huge performance enhancement while significantly reducing the hardware requirements.

The key contributions of our work are summarized as follows:

(1) For the light-weight upgrade, based on the light-weight characteristics of FasterNet and C3, a C3Fast module is proposed to replace the C3 module in the backbone.
(2) For the detection accuracy enhancement, the SIoU is adopted to replace the original loss module. The concept of the vector angle and the redefined distance loss are fully utilized to effectively accelerate the convergence of the network and improve the detection performance of the model. Moreover, the SA with the shuffle and reorganization mechanism is integrated into the neck part, which can effectively improve the network feature expression and extract the important information in the image.
(3) The data enhancement method is used to process the collected fastener screw images, including randomly flipped, cropped, and scaled at random. Meanwhile, the parameters, such as the noise color and brightness, are adjusted appropriately to establish datasets with different angles, different brightness, and different noise. Thus, the real-time change in the environmental conditions in real situations can be truly simulated, which enriches the diversity of the datasets and strengthens the recognition ability of the algorithm under complex weather conditions.
(4) Compared with the baseline model, our work can reduce by 17.4% the Params and 19.5% the FLOPs, while the P, mAP@50, Recall, and F1 scores are improved by 10.6%, 6.4%, 13.4%, and 12.2%.

The rest of the paper is organized as follows. Section 2 describes the theories related to deep learning; Section 3 describes the framework of the baseline and improvement implementation of the proposed model; Section 4 introduces the dataset and experimental environment; Section 5 conducts the experimental comparative analysis and verifies the effectiveness of the algorithm; and the full text work and future research focus are summarized in Section 6.

2. Related Work

2.1. YOLOv5

Compared with earlier YOLO series methods, YOLOv5 significantly improves the computation speed while the accuracy remains high. The feature extraction procedure is handled by the backbone, the feature fusion step is managed by the neck, and the detection task is performed by the head module.

The backbone module is formed by CBS, C3, and SPPF. CBS is designed for preliminary feature extraction, which contains Conv [24], BatchNorm [25], and SiLU [26]. The C3 module is the main feature extraction structure of YOLOv5, which can achieve features from feature maps through the convolution process. This structure can be replaced with convolution nerve networks like RestNet [27], CSPNet [28], ShuffleNet [29], MobileNet [30], GhostNet [31], and FasterNet [32] to reduce the computation cost. The function of SPPF is for the multi-scale feature fusion.

The neck module of YOLOv5 is a feature pyramid network (FPN) [33] that can fuse low-level features and high-level features of the input image to achieve comprehensive features, improving the accuracy and robustness of detection. It can be replaced with other FPNs, like PAN [34] and BiFPN [35]. The head module converts feature maps into detection boxes and category probabilities to detect target categories and positions.

2.2. IOU

The intersection over union (IoU) is an algorithm of the degree of overlapping between the prediction box and the ground truth box. For the YOLO series, CIoU [36], GIoU [37], DIoU [38], and SIoU [39] are commonly used for performance improvement. GIoU focuses not only on overlapping regions but also on other non-overlapping regions. It presents the degree of overlapping better than the original IoU. When one box is inside the other box, GIoU will degrade to IoU. To deal with this problem, DIoU is introduced, which takes the distance, overlapping rate, and scale between the anchor box and the bounding box into account, accelerating the bounding box regression. But when both boxes' central point coincides, DIoU becomes IoU. CIoU introduces the length–width ratio into the loss function based on DIoU, yet the actual length and width are not considered. SIoU can rapidly shift the predicted box toward the nearest axis, allowing regression for only one coordinate (X or Y), greatly increasing the computation speed through effectively reducing the time cost in the convergence process.

2.3. Attention Model

The attention mechanism plays an important role in the field of computer vision. Significant recognition accuracy promotion can be achieved by incorporating an attention module into the critical feature extraction process, the mechanism of which is to focus on the critical information of the current task and reduce the weight factor of irrelevant information. This approach can improve the performance of the model, allowing it to handle complex tasks more efficiently. There are many kinds of attention mechanisms, such as CBAM [40], SENet [41] and SimAM [42]. We use shuffle attention (SA) [43], which can significantly improve the model performance through a clever combination of the spatial domain and channel domain. SA divides the channel features to obtain multiple groups of sub-features and then uses the Shuffle Unit to process each sub-feature from the spatial and channel aspects, respectively. Finally, all the processed sub-features are gathered, and then a Channel Shuffle operation is used to fuse different groups of sub-features, which can significantly improve model performance.

3. Methods

3.1. Proposed Model

We propose the FSS-YOLO model based on the YOLOv5n model, and the architecture is shown in Figure 2. The network structure of YOLOv5n consists of several CBS and C3 modules. The proposed model replaces all the C3 modules with C3Fast. The loss function for the bounding box regression is SIoU, instead of the original IoU. For the FPN in the neck module, SA is added into the bottom-up process, increasing the model's attention during the up-sample pathway. Adopting these methods can significantly increase the model's efficiency and accuracy.

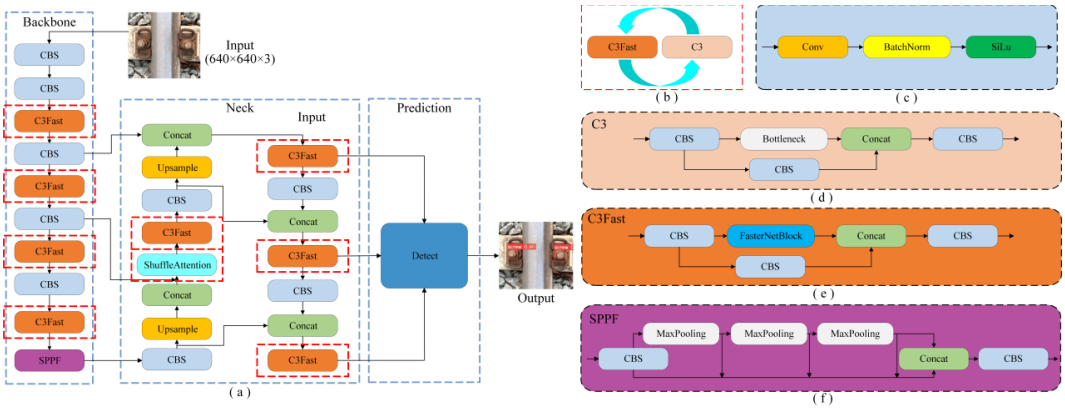

Figure 2. Scheme of FSS-YOLO: (**a**) structure of the proposed model; (**b**) the light-weight module replacement; (**c**) CBS module; (**d**) C3 module; (**e**) C3Fast module; and (**f**) SPPF module.

3.2. FasterNet

In order to achieve a higher inference speed, Chen et al. proposed FasterNet [32] (Figure 3) based on PConv [44]. It is a light-weight model that significantly enhances efficiency and retains better performance. This module can considerably reduce the weight of the network, while cutting down the computational redundancy and memory access frequency. There are four stages in the top part of the FasterNet model, and each stage has a FasterNet Block and an embedding/merging layer. The bottom three layers form a classifier. In the FasterNet Block, a PConv layer is followed by two PWConv layers. Only a batch normalization (BN) layer and an activity layer are placed between the PWConv layers aiming to retain the feature diversity and achieve a faster speed. PConv undertakes feature extraction through partial input channels. Thus, the FLOPs of PConv are:

$$h \times w \times k^2 \times c_p^2 \tag{1}$$

where h is the height of PConv, w is the width of PConv, and k is the height and width of the filter. The c_p is set as $^1/_4 c$, and PConv's FLOPs are only $^1/_{16}$ of normal Conv. Thus, the PConv's memory access is much smaller than that of regular convolution.

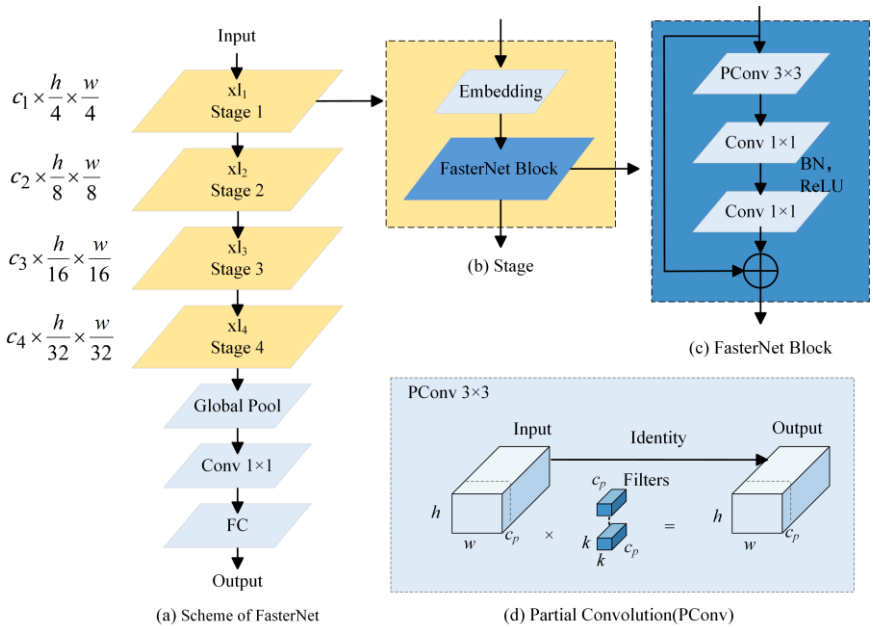

Figure 3. Scheme of FasterNet: (**a**) FasterNet structure; (**b**) schematic diagram of each stage; (**c**) schematic diagram of the FasterNet Block; and (**d**) schematic diagram of the PConv mechanism.

3.3. SIoU

The IoU schematic is shown in Figure 4. The score of the IoU increases as the intersection between the ground truth box and the prediction box increases. This situation indicates high accuracy, and vice versa.

The IoU is obtained through the following equation:

$$IoU = \left| \frac{A \cap B}{A \cup B} \right| \tag{2}$$

A represents the ground truth box, while B represents the prediction box. The corresponding IoU loss function is:

$$L_{IoU} = 1 - IoU \tag{3}$$

Our work adopts the SIoU [39] as a loss function for accuracy promotion. This method defines the prediction loss from three aspects: angle cost, distance cost, and shape cost.

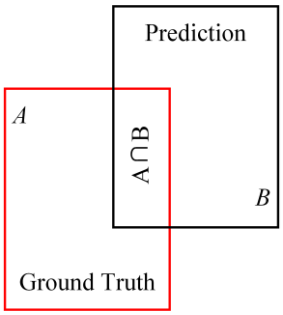

Figure 4. Schematic diagram of the IoU.

3.3.1. Angle Cost

This cost is introduced to reduce the variables in the distance calculation. As Figure 5 shows, the model brings the prediction box to the nearest axis and then moves to the ground truth box along this axis.

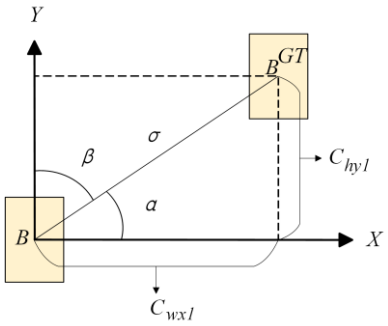

Figure 5. The angle cost.

The angle cost Λ is calculated from the following equation (when the nearest axis is the x axis):

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

where the angle x is:

$$x = \frac{c_{hy1}}{\sigma} = \sin(\alpha) \tag{5}$$

c_{hy1} and σ are the centroid point offset in height and distance between the ground truth box and the prediction box:

$$\sigma = \sqrt{\left(b_{c_x}^{gt} - b_{c_x}\right)^2 + \left(b_{c_y}^{gt} - b_{c_y}\right)^2} \tag{6}$$

$$c_{hy1} = \max\left(b_{c_x}^{gt} - b_{c_y}\right) - \min\left(b_{c_y}^{gt} - b_{c_y}\right) \tag{7}$$

where $\left(b_{c_x}^{gt}, b_{c_y}^{gt}\right)$ and $\left(b_{c_x}, b_{c_y}\right)$ represent the centroid point coordinates of the ground truth box and prediction box.

3.3.2. Distance Cost

The distance cost Δ is designed according to the angle cost:

$$\Delta = \sum_{t=x,y} \left(1 - e^{-\gamma \rho_t}\right) \tag{8}$$

$$\rho_x = \left(\frac{b_{c_x}^{gt} - b_{c_x}}{c_{wx2}}\right)^2, \rho_y = \left(\frac{b_{c_y}^{gt} - b_{c_y}}{c_{hy2}}\right)^2, \gamma = 2 - \Lambda \tag{9}$$

As Figure 6 shows, c_{wx2} and c_{hy2} are the vertical and horizontal parameters of the minimum bounding rectangle based on the ground truth box and prediction box. As the value of angle α moves from 0 to $\pi/4$, the distance cost grows dramatically.

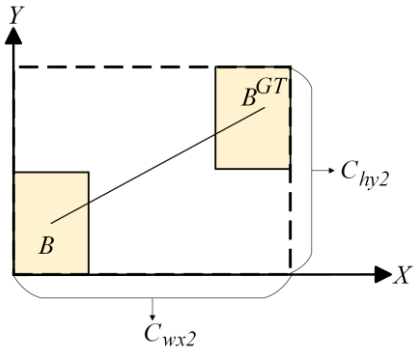

Figure 6. The distance cost.

3.3.3. Shape Cost

The function of the shape cost is:

$$\Omega = \sum \left(1 - e^{-wt}\right)^{\theta} \tag{10}$$

$$w_w = \frac{\left|w^p - w^{gt}\right|}{\max(w^p, w^{gt})}, \quad w_h = \frac{\left|h^p - h^{gt}\right|}{\max(h^p, h^{gt})} \tag{11}$$

in which (w^p, h^p) and (w^{gt}, h^{gt}) represent the actual vertical and horizontal parameters of the ground truth box and prediction box. The θ is a critical parameter that is unique for a specific case.

3.4. Attention

The channel attention mechanism can enhance the weighting factor of important channels, while the spatial attention mechanism can highlight important information in the feature map. However, using these two mechanisms at the same time will increase the computational cost of the model. To deal with this problem, the shuffle attention (SA) is proposed by Zhang et al. [43], with the structure shown in Figure 7. The SA combines the advantages of channel-wise attention and spatial-wise attention, while maintaining the efficiency of the network. We introduce this light-weight attention mechanism module for the higher feature extraction ability. The SA separates the input feature maps $X \in R^{C \times H \times W}$ into G groups, which is $X = [X_1, \ldots, X_G]$, $X_k \in R^{C/G \times H \times W}$. Each group splits the channel and spatial attention pathways, i.e., X_{k1}, $X_{k2} \in R^{C/2G \times H \times W}$, and the outputs of both attention mechanisms are:

$$X'_{k1} = \sigma(\mathcal{F}_c(s)) \cdot X_{k1} = \sigma(W_1 s + b_1) \cdot X_{k1} \tag{12}$$

$$X'_{k2} = \sigma(W_2 \cdot GN(X_{k2}) + b_2) \cdot X_{k2} \tag{13}$$

where s indicates the global averaging pooling, GN is the group norm [45], and W_1, b_1, W_2, and b_2 are within $R^{C/2G \times 1 \times 1}$. The results of the attention process are connected. The whole G groups are processed parallelly and fused with the channel shuffle operator for communication between each group. Note that the SA outputs are the same size as input X, making it convenient for CNN integration.

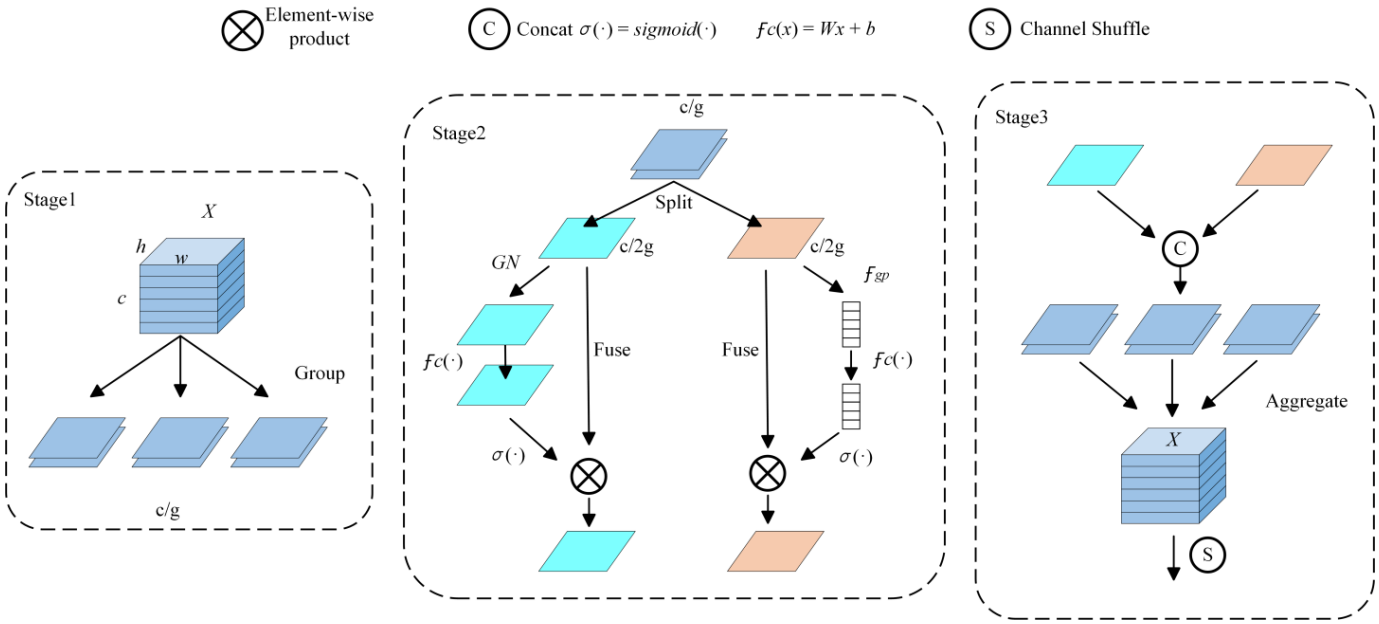

Figure 7. Scheme of the shuffle attention.

4. Experiment

4.1. Dataset

The fastener screw detection performance of the fastener screw maintenance robot may be significantly interfered with when encountering unexpected environmental factors. For example, the hexagonal contour of the fastener screw bears a resemblance to a few ballast contours, making it challenging to differentiate between them against complex environmental backgrounds. Additionally, foreground occlusions from other trash may obstruct the complete exposure of the fastener screw, further complicating detection tasks.

Due to the datasets on the railway being predominately based on the defects of the railway system, the lack of a suitable fastener screw dataset hinders our research. To enhance the generalization capability of the FSS-YOLO, a fastener dataset that contains a broad range of environmental conditions, i.e., sunny, snowy, mottled sun, rainy, and occluded circumstances, is presented. The data enhancement methods involve flipping, cropping, grayscale, color adjustment, brightness adjustment, noise adjustment, and sharpening, expanding the dataset from the original 250 images to 1000 images.

4.2. Experiment Setup

Finally, the configuration of the dataset is shown in Table 1 to ensure a comprehensive evaluation of the models tested in the experiment.

Table 1. Dataset.

Dataset	Scale
Test	1
Train	8
Val	1

The experiment's parameters and the environment adopted are listed in Tables 2 and 3.

Table 2. Experimental environment: Parameter configuration.

Configuration	Version
System	Windows 11
CPU	Intel Core i5-12500H @3.10 GHz
GPU	NVIDIA GeForce RTX 3060
Language	Python 3.8
Acceleration	CUDA 12.0
Framework	PyTorch 1.13.0

Table 3. Parameter configuration.

Parameters	Details
Optimization algorithm	SGD
Learning rate	0.01
Batch size	8
Img size	640 × 640
Epochs	100

4.3. Evaluation Indicators

We utilize the parameters (Params), floating-point operations per second (FLOPs), precision (P), mean average precision (mAP@50), Recall(R), F1 score, and frames per second (FPS) to assess the model's performance. The high mAP@50, Recall, and F1 scores denote the accuracy of the tested model. The miniature Params, FLOPs, and a considerable FPS suggest a more satisfactory real-time model to have been presented. The formula is as follows:

$$Precision = \frac{TP}{TP + FP} \tag{14}$$

$$Recall = \frac{TP}{TP + FN} \tag{15}$$

where TP represents the count of correctly predicted positive class samples, FP represents the count of negative samples incorrectly predicted as positive, and FN represents the count of positive samples wrongly predicted as negative.

In general, mAP@50 is used as the most important evaluation criteria for models, and its calculation functions are as follows:

$$AP = \int_0^1 P(R)dR \tag{16}$$

$$mAP = \frac{1}{N}\sum_{i=1}^{N} AP_i \tag{17}$$

where the N denotes the total number of categories in the detection task. For this study, the N is set as 1, $AP = mAP$.

5. Discussion

To satisfy the fast detection requirement of the fastener screw maintenance robot, we choose the YOLOv5n model with the smallest Params and lowest computation cost as the baseline. Although YOLOv5m and YOLOv5s perform well in terms of the P, mAP, Recall, and F1 scores, their network Params and FLOPs are too large, resulting in a huge computation resource expenditure during operation. This means that these two models are too slow to complete the real-time recognition job on devices with low capability.

Then, different IoU loss functions are adopted, and the results are shown in Table 4. It shows that the SIoU loss function can achieve the best FPS with higher overall performance. Therefore, the YOLOv5n with SIoU is used.

Table 4. Comparison results of loss functions with the same baseline.

Model	Params	FLOPs (G)	P	mAP@50	Recall	F1 Score	FPS
YOLOv5n+CIoU	1760518	4.1	0.819	0.635	0.556	0.662	176
YOLOv5n+DIoU	1760518	4.1	0.942	0.675	0.534	0.681	184
YOLOv5n+GIoU	1760518	4.1	**0.956**	0.712	0.535	0.686	186
YOLOv5n+SIoU	1760518	4.1	0.917	**0.752**	**0.623**	**0.741**	**192**

Table 5 lists the light-weight modifications using MobileNetv3, GhostNet, and C3Fast. The Params and FLOPs are significantly reduced, but the P, mAP@50, Recall, and F1 scores are decreased simultaneously. The C3Fast achieves the best score within the three modules, while its Params and FLOPs are acceptable. A great enhancement in the FPS appears due to the hardware latency optimization ability of the FasterNet Block-improved C3Fast module. Therefore, C3Fast is used for the light-weight upgradation. For higher detection accuracy, an attention mechanism is introduced. As Table 6 shows, barely any difference appears between the Params and FLOPs. The SA wins in terms of the P, mAP@50, and F1, while there is only a slight lack in Recall. The combination of channel-wise attention and spatial-wise attention has superior performance in this fastener screw detection task. Thus, the adopted modules for performance enhancement are the C3Fast, SIoU, and SA.

Table 5. Quantitative indicators of different light-weight models.

Model	Params	FLOPs (G)	P	mAP@50	Recall	F1 Score	FPS
YOLOv5n	1760518	4.1	0.819	0.635	0.556	0.662	176
YOLOv5n+MobileNetv3	**792044**	**1.2**	0.655	0.318	0.296	0.407	137
YOLOv5n+GhostNet	1283706	2.9	0.737	0.413	0.377	0.498	148
YOLOv5n+C3Fast	1452886	3.3	**0.76**	**0.466**	**0.418**	**0.539**	**199**

Table 6. Comparison of the performance of various attention mechanisms.

Model	Params	FLOPs (G)	P	mAP@50	Recall	F1 Score	FPS
YOLOv5n+C3Fast+SIoU+SEAttention	1461078	3.4	0.853	0.624	0.573	0.685	166
YOLOv5n+C3Fast+SIoU+SimAM	**1452886**	**3.3**	0.803	0.59	**0.634**	0.708	185
YOLOv5n+C3Fast+SIoU+SA	1452982	**3.3**	**0.906**	**0.676**	0.631	**0.743**	**189**

We compare the metrics for each modification stage, and the results are presented in Table 7. C3Fast significantly improves the computational speed, but the P, mAP@50, Recall, and F1 scores appear to diminish. Then, SIoU and SA are adopted to compensate for the loss in the P, mAP@50, Recall, and F1 scores.

Table 7. Indicator comparison before and after modification.

Model	Params	FLOPs (G)	P	mAP@50	Recall	F1 Score	FPS
YOLOv5n	1760518	4.1	0.819	0.635	0.556	0.662	176
YOLOv5n+C3Fast	**1452886**	**3.3**	0.76	0.466	0.418	0.539	**199**
YOLOv5n+C3Fast+SIoU	**1452886**	**3.3**	0.891	0.644	0.537	0.67	184
YOLOv5n+C3Fast+SIoU+SA	1452982	**3.3**	**0.906**	**0.676**	**0.631**	**0.743**	189

To quantitatively evaluate the overall performance of the FSS-YOLO, several light-weight YOLO series models are trained and tested on fastener screw datasets. As shown in Table 8, FSS-YOLO exhibits significantly improved performance in all the aspects compared to the baseline. The mAP@50 of FSS-YOLO is 14.5% higher than that of YOLOv7-Tiny and the F1 score is 13% higher. The mAP@50 of FSS-YOLO is 7.1% higher than that of YOLOv8 and the F1 score is 3.7% higher. In general, compared with YOLOv5n, YOLOv7-tiny, and YOLOv8n, our work's performance is the best. The FSS-YOLO not only has the smallest Params and FLOPs but also improves the detection accuracy, which means that it is more conducive to completing the detection task of the fastener screw maintenance robot. The actual test results are shown in Figure 8.

Table 8. Quantitative indicators of the YOLO series.

Model	Params	FLOPs (G)	P	mAP@50	Recall	F1 Score	FPS
YOLOv5n	1760518	4.1	0.819	0.635	0.556	0.662	176
YOLOv7-tiny	6007596	13.0	0.859	0.59	0.533	0.657	170
YOLOv8n	3005843	8.1	0.829	0.631	**0.675**	0.716	166
FSS-YOLO (Ours)	**1452982**	**3.3**	**0.906**	**0.676**	0.631	**0.743**	**189**

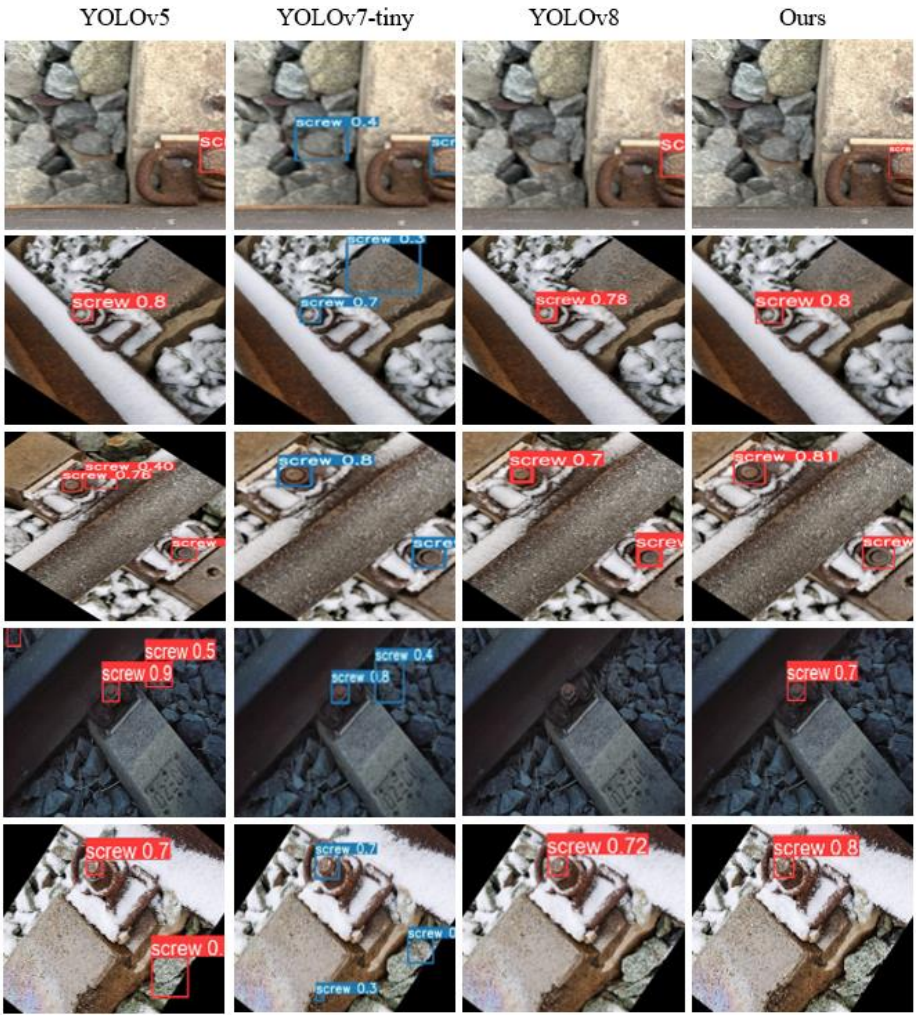

Figure 8. The results are compared with those of other algorithms.

6. Conclusions

Computer vision technology based on deep learning has been widely used in the industrial field. Although existing algorithms can meet the basic needs of many practical applications, their performance can be further improved by optimizing existing models. Based on the analysis of the current competitive one-stage detection algorithms (YOLOv5, YOLOv7 and YOLOv8), this paper focuses on the network structure upgrade of the YOLOv5, which has good accuracy and fast inference speed.

We propose the FSS-YOLO model, which aims to enhance the real-time inspection capability of a fastener screw maintenance robot, which can only offer limited computing resources. The C3Fast module is proposed to substitute for the C3 module in the backbone for the model's efficiency promotion. This modification results in the computation speed increasing while the P reduces. In our approach, the SIoU instead of the original CIoU is utilized. The SIoU introduces the concept of the vector angle and redefines the distance loss, which effectively reduces the degrees of freedom in the regression. This leads to the faster convergence of the network and improves the regression accuracy, allowing the

model to efficiently identify fastener screws. Additionally, we integrate the SA attention mechanism (Shuffle Attention) in the sampling process of the neck network. The shuffling and reorganization of input features effectively enhances the feature expression ability of fastener screws.

Aiming at addressing the issue of insufficient datasets of fastener screws, a new dataset of fastener screws based on various environments is built. The data enhancement method is used to expand the samples, and images with different processing degrees are randomly selected to simulate the images collected by the robot in different weathers, which can effectively avoid the similarity of the image features and improve the generalization ability of the model.

The experiment result shows that our work is 7.3% faster in terms of the FPS than the baseline model, while reducing by 17.4% and 19.5% the Params and FLOPs, and the P, mAP@50, Recall, and F1 scores are improved by 10.6%, 6.4%, 13.4%, and 12.2%. Extensive evaluations demonstrate the superiority of the proposed model when compared with some competitive models. Meanwhile, the fastener screw maintenance robot obtains a substantial advancement in its competency for the fastener screw detection task. At present, the fastener screw maintenance robot developed based on FSS-YOLO has conducted more than 200 practical experiments and is about to be used for disassembly and assembly tasks concerning type II fasteners. The performance of this robot is satisfactory.

In the future, we will conduct in-depth research on the adaptability of the model to improve the recognition accuracy and reduce the false detection rate in more complex actual environments, like dim conditions and partly covered by trash, so that the model can perform better, thereby improving the fastener screw detection function of the fastener screw maintenance robot.

Author Contributions: Conceptualization, Y.C., F.Z. and M.H.; methodology, Y.C., H.Z., Q.T. and M.H.; software, Y.C., J.X. and M.H.; writing—original draft, M.H.; writing—review and editing, Y.C., J.X., Q.T. and H.Z.; funding acquisition, Y.C., H.Z. and F.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Natural Science Foundation of Hubei Province (2022CFB882), Knowledge Innovation Project of Wuhan Science and Technology Bureau (2023010201020373), Key Research and Development Program of Hubei Province of China (2023BAB088), and the High-Level Talent Fund of Hubei University of Technology (BSQD2020010).

Data Availability Statement: The raw data supporting the conclusions of this article will be made available by the authors on request.

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

References

1. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23 June 2014; pp. 580–587.
2. He, K.; Gkioxari, G.; Dollar, P.; Girshick, R. Mask R-CNN. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 2980–2988.
3. Girshick, R. Fast R-CNN. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
4. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1137–1149. [CrossRef] [PubMed]
5. Shuang, F.; Wei, S.; Li, Y.; Gu, X.; Lu, Z. Detail R-CNN: Insulator Detection Based on Detail Feature Enhancement and Metric Learning. *IEEE Trans. Instrum. Meas.* **2023**, *72*, 2524414. [CrossRef]
6. Wei, X.; Yang, Z.; Liu, Y.; Wei, D.; Jia, L.; Li, Y. Railway Track Fastener Defect Detection Based on Image Processing and Deep Learning Techniques: A Comparative Study. *Eng. Appl. Artif. Intell.* **2019**, *80*, 66–81. [CrossRef]
7. He, Q.; Liu, J.; Huang, Z. WSRC: Weakly Supervised Faster RCNN Toward Accurate Traffic Object Detection. *IEEE Access* **2023**, *11*, 1445–1455. [CrossRef]
8. Yu, Y.; Zhang, K.; Yang, L.; Zhang, D. Fruit Detection for Strawberry Harvesting Robot in Non-Structural Environment Based on Mask-RCNN. *Comput. Electron. Agric.* **2019**, *163*, 104846. [CrossRef]

9. Hong, S.; Jiang, Z.; Liu, L.; Wang, J.; Zhou, L.; Xu, J. Improved Mask R-CNN Combined with Otsu Preprocessing for Rice Panicle Detection and Segmentation. *Appl. Sci.* **2022**, *12*, 11701. [CrossRef]
10. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
11. Redmon, J.; Farhadi, A. YOLO9000: Better, Faster, Stronger. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 6517–6525.
12. Redmon, J.; Farhadi, A. YOLOv3: An Incremental Improvement. *arXiv* **2018**, arXiv:1804.02767.
13. Bochkovskiy, A.; Wang, C.-Y.; Liao, H.-Y.M. YOLOv4: Optimal Speed and Accuracy of Object Detection. *arXiv* **2020**, arXiv:2004.10934.
14. Zhao, X.; Xiao, N.; Cai, Z.; Xin, S. YOLOv5-Sewer: Lightweight Sewer Defect Detection Model. *Appl. Sci.* **2024**, *14*, 1869. [CrossRef]
15. Mushtaq, F.; Ramesh, K.; Deshmukh, S.; Ray, T.; Parimi, C.; Tandon, P.; Jha, P.K. Nuts&bolts: YOLO-v5 and Image Processing Based Component Identification System. *Eng. Appl. Artif. Intell.* **2023**, *118*, 105665. [CrossRef]
16. Zhuang, L.; Qi, H.; Wang, T.; Zhang, Z. A Deep-Learning-Powered Near-Real-Time Detection of Railway Track Major Components: A Two-Stage Computer-Vision-Based Method. *IEEE Internet Things J.* **2022**, *9*, 18806–18816. [CrossRef]
17. He, H. Automatic Assembly of Bolts and Nuts Based on Machine Vision Recognition. *J. Phys. Conf. Ser.* **2021**, *2113*, 012033. [CrossRef]
18. Panigrahi, S.; Raju, U.S.N. DSM-IDM-YOLO: Depth-Wise Separable Module and Inception Depth-Wise Module Based YOLO for Pedestrian Detection. *Int. J. Artif. Intell. Tools* **2023**, *32*, 2350011. [CrossRef]
19. Ma, Y.; Chai, L.; Jin, L.; Yu, Y.; Yan, J. AVS-YOLO: Object Detection in Aerial Visual Scene. *Int. J. Patt. Recogn. Artif. Intell.* **2022**, *36*, 2250004. [CrossRef]
20. Jiang, B.; Qu, R.; Li, Y.; Li, C. VC-YOLO: Towards Real-Time Object Detection in Aerial Images. *J. Circuit. Syst. Comp.* **2022**, *31*, 2250147. [CrossRef]
21. Chen, T.; Ding, Z.; Li, B. Elderly Fall Detection Based on Improved YOLOv5s Network. *IEEE Access* **2022**, *10*, 91273–91282. [CrossRef]
22. Liu, S.; Wang, Y.; Yu, Q.; Liu, H.; Peng, Z. CEAM-YOLOv7: Improved YOLOv7 Based on Channel Expansion and Attention Mechanism for Driver Distraction Behavior Detection. *IEEE Access* **2022**, *10*, 129116–129124. [CrossRef]
23. Wang, Y.; Wang, H.; Xin, Z. Efficient Detection Model of Steel Strip Surface Defects Based on YOLO-V7. *IEEE Access* **2022**, *10*, 133936–133944. [CrossRef]
24. Sunkara, R.; Luo, T. No More Strided Convolutions or Pooling: A New CNN Building Block for Low-Resolution Images and Small Objects. *arXiv* **2022**, arXiv:2208.03641.
25. Ioffe, S.; Szegedy, C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *arXiv* **2015**, arXiv:1502.03167.
26. Elfwing, S.; Uchibe, E.; Doya, K. Sigmoid-Weighted Linear Units for Neural Network Function Approximation in Reinforcement Learning. *Neural Netw.* **2018**, *107*, 3–11. [CrossRef] [PubMed]
27. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
28. Wang, C.-Y.; Mark Liao, H.-Y.; Wu, Y.-H.; Chen, P.-Y.; Hsieh, J.-W.; Yeh, I.-H. CSPNet: A New Backbone That Can Enhance Learning Capability of CNN. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Seattle, WA, USA, 14–19 June 2020; pp. 1571–1580.
29. Zhang, X.; Zhou, X.; Lin, M.; Sun, J. ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 6848–6856.
30. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *arXiv* **2017**, arXiv:1704.04861.
31. Han, K.; Wang, Y.; Tian, Q.; Guo, J.; Xu, C.; Xu, C. GhostNet: More Features from Cheap Operations. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 1577–1586.
32. Chen, J.; Kao, S.; He, H.; Zhuo, W.; Wen, S.; Lee, C.-H.; Chan, S.-H.G. Run, Don't Walk: Chasing Higher FLOPS for Faster Neural Networks. In Proceedings of the 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, 17–24 June 2023; pp. 12021–12031.
33. Lin, T.-Y.; Dollar, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature Pyramid Networks for Object Detection. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 936–944.
34. Liu, S.; Qi, L.; Qin, H.; Shi, J.; Jia, J. Path Aggregation Network for Instance Segmentation. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 8759–8768.
35. Zhu, L.; Deng, Z.; Hu, X.; Fu, C.-W.; Xu, X.; Qin, J.; Heng, P.-A. Bidirectional Feature Pyramid Network with Recurrent Attention Residual Modules for Shadow Detection. In *Computer Vision—ECCV 2018*; Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y., Eds.; Lecture Notes in Computer Science; Springer International Publishing: Cham, Switzerland, 2018; Volume 11210, pp. 122–137, ISBN 978-3-030-01230-4.

36. 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]
37. Rezatofighi, H.; Tsoi, N.; Gwak, J.; Sadeghian, A.; Reid, I.; Savarese, S. Generalized Intersection Over Union: A Metric and a Loss for Bounding Box Regression. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 658–666.
38. Zheng, Z.; Wang, P.; Liu, W.; Li, J.; Ye, R.; Ren, D. Distance-IoU Loss: Faster and Better Learning for Bounding Box Regression. *AAAI* **2020**, *34*, 12993–13000. [CrossRef]
39. Gevorgyan, Z. SIoU Loss: More Powerful Learning for Bounding Box Regression. *arXiv* **2022**, arXiv:2205.12740.
40. Woo, S.; Park, J.; Lee, J.-Y.; Kweon, I.S. CBAM: Convolutional Block Attention Module. In Proceedings of the European conference on computer vision (ECCV), Munich, Germany, 8–14 September 2018.
41. Hu, J.; Shen, L.; Sun, G. Squeeze-and-Excitation Networks. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7132–7141.
42. Yang, L.; Zhang, R.-Y.; Li, L.; Xie, X. SimAM: A Simple, Parameter-Free Attention Module for Convolutional Neural Networks. In Proceedings of the International Conference on Machine Learning, Virtual, 18–24 July 2021.
43. Zhang, Q.-L.; Yang, Y.-B. SA-Net: Shuffle Attention for Deep Convolutional Neural Networks. In Proceedings of the ICASSP 2021—2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Toronto, ON, Canada, 6 June 2021; pp. 2235–2239.
44. Ma, X.; Guo, F.-M.; Niu, W.; Lin, X.; Tang, J.; Ma, K.; Ren, B.; Wang, Y. PCONV: The Missing but Desirable Sparsity in DNN Weight Pruning for Real-Time Execution on Mobile Devices. *AAAI* **2020**, *34*, 5117–5124. [CrossRef]
45. Wu, Y.; He, K. Group Normalization. In *Computer Vision—ECCV 2018*; Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y., Eds.; Lecture Notes in Computer Science; Springer International Publishing: Cham, Switzerland, 2018; Volume 11217, pp. 3–19, ISBN 978-3-030-01260-1.