applied sciences

Article

A Lightweight Method for Detecting Sewer Defects Based on Improved YOLOv5

Xing Zhang [1], Jiawei Zhang [2], Lei Tian [2], Xiang Liu [2,*] and Shuohong Wang [3]

[1] School of Management, Shanghai University of Engineering Science, Shanghai 201620, China
[2] School of Electronic and Electric Engineering, Shanghai University of Engineering Science, Shanghai 201620, China
[3] Department of Molecular and Cellular Biology and Center for Brain Science, Harvard University, Cambridge, MA 02138, USA
* Correspondence: xliu@sues.edu.cn

Abstract: In response to the issues of the existing sewer defect detection models, which are not applicable to small computing platforms due to their complex structure and large computational volume, as well as the low detection accuracy, a lightweight detection model based on YOLOv5, named YOLOv5-GBC, is proposed. Firstly, to address the computational redundancy problem of the traditional convolutional approach, GhostNet, which is composed of Ghost modules, is used to replace the original backbone network. Secondly, aiming at the problem of low detection accuracy of small defects, more detailed spatial information is introduced by fusing shallow features in the neck network, and weighted feature fusion is used to improve the feature fusion efficiency. Finally, to improve the sensitivity of the model to key feature information, the coordinate attention mechanism is introduced into the Ghost module and replaced the traditional convolution approach in the neck network. Experimental results show that compared with the YOLOv5 model, the model size and floating point of operations (FLOPs) of YOLOv5-GBC are reduced by 74.01% and 74.78%, respectively; the mean average precision (MAP) and recall are improved by 0.88% and 1.51%, respectively; the detection speed is increased by 63.64%; and the model size and computational volume are significantly reduced under the premise of ensuring the detection accuracy, which can effectively meet the needs of sewer defect detection on small computing platforms.

Keywords: sewer defect detection; YOLOv5; GhostNet; lightweight; coordinate attention mechanism

check for updates

Citation: Zhang, X.; Zhang, J.; Tian, L.; Liu, X.; Wang, S. A Lightweight Method for Detecting Sewer Defects Based on Improved YOLOv5. *Appl. Sci.* **2023**, *13*, 8986. https://doi.org/10.3390/app13158986

Academic Editor: Andrea Prati

Received: 13 June 2023
Revised: 31 July 2023
Accepted: 1 August 2023
Published: 5 August 2023

1. Introduction

Due to the long-term burial of drainage pipeline systems underground, it is inconvenient to detect and mitigate the various structural or functional defects that exist. These defects are one of the most direct threats to urban safety. These defects not only cause problems such as sewer blockage and leakage, but also serious safety accidents [1], potentially resulting in ground collapse and environmental pollution occurring at the same time. Therefore, it is of great importance for urban safety to discover existing defects accurately and quickly.

The widely used defect detection method for drainage sewers is the Closed-Circuit Television (CCTV) inspection technology, which is mainly divided into two stages: the field stage and the office stage [2]. During the field stage, the technician controls a robot with a camera to shoot a detailed record of the internal conditions of the sewer and saves the video data to a storage device for the technician to review during the office stage. In the office stage, the technician scrutinizes the video frames of the sewers' defects. This method mainly relies on manual interpretation by technicians, which requires a lot of manpower and has high labor intensity. The manual interpretation is prone to visual fatigue and misjudgment, and requires a high level of professional expertise, which may lead to compliance issues when demands for sewer inspections continue to increase. For instance, a drainage group in Beijing city needs to inspect about 1500 to 2000 km of sewers every year and watch 15,000 G to 20,000 G of assessment video, on average [3]. Therefore, there is an urgent need to develop an automated method for detecting sewer defects.

With the development of deep learning technology, object detection technology that can perform both classification and detection is widely used for vehicle recognition [4], hand script counterfeit detection [5], and automatic detection of drainage sewers in China and abroad [6,7]. The most commonly used methods include Single Shot Multibox Detector (SSD) [8], You Only Look Once (YOLO) [9], and Faster RCNN [10], which effectively improve the detection accuracy by using neural networks to automatically extract features. For example, Yin et al. [11] used the YOLOv3 model to construct a sewer defect automatic detection system, achieving efficient detection of six types of defects such as cracks. Cheng et al. [12] used the Faster R-CNN algorithm to detect defects such as tree roots and sediment in CCTV videos and studied the influence of different network structures and dataset sizes on model performance. Wang Huanhuan et al. [3] used the YOLOv5 algorithm for automatic detection of six kinds of defects including blockages and constructed a rating system for the sewer status, proving the feasibility of the YOLOv5 model in sewer defect detection tasks. Li et al. [13] used an improved Faster R-CNN model to fuse global contextual features with local defect features for sewer defect localization and fine-grained classification.

The focus of the above-mentioned method aims to improve the detection accuracy of the model, which also leads to increasingly complex model structures and higher hardware requirements. In actual detection operations, the terminal control device operated by field personnel is usually a low-performance industrial computer, with limited computational resources such as graphics cards and memory. This makes it impossible to deploy complex and computationally intensive detection models, thereby failing to meet the real-time detection needs of sewer defects. In actual sewer defect detection tasks, due to the complex and numerous sewers, a large number of detection videos are produced, while professional detection personnel are very limited [14]. If the office technicians cannot judge the videos in a timely manner, maintenance of serious defects may be delayed; thus, researching lightweight models that are suitable for small mobile devices is also of great significance.

The single-stage object detection networks, which avoid the drawbacks of two-stage detection networks based on candidate boxes such as slower detection speed, are more suitable for applications that require rapid detection. Among them, YOLOv5 has previously added various improvement strategies to the YOLO series of networks, and can achieve a balance between higher detection accuracy and detection speed, making it the best choice for pipeline defect detection tasks. Therefore, this paper proposes a lightweight model for sewer defect detection, YOLOv5-GBC, by improving the YOLOv5 model. The specific improvements are as follows:

(1) By replacing the backbone network of YOLOv5 with a lightweight network called GhostNet, which is suitable for mobile platforms, the problem of calculating redundancy in traditional convolution methods is solved, greatly reducing the volume and computing power of the model.

(2) A new feature fusion network is proposed, which improves the neck network based on the weighted feature fusion and same-scale feature residual connection in the Bidirectional Feature Pyramid Network (BiFPN) [15]. Shallow feature increase is introduced to provide the model with detailed information in images, enhancing the efficiency of feature fusion and the accuracy of detecting tiny defects.

(3) By introducing a coordinate attention mechanism [16] to improve the bottleneck modules of the neck network, the model can be made lightweight while achieving higher sensitivity to critical features.

2. Lightweight YOLOv5 Sewer Defect Detection Model

According to the different depths and widths of the network, YOLOv5 can be divided into four different models, named s, m, l, and x according to their size. The larger models usually achieve better detection results in various computer vision tasks but require grater hardware resources. This means they are unsuitable for deployment on small mobile devices and unable to meet the real-time detection requirements in the field. Therefore, this paper chooses the YOLOv5 model as the main research subject, which has the moderate model size and detection speed suitable for the practical requirements of pipeline defect detection.

The sewer defect detection task faces difficulties such as high workload, low defect image resolution, and limited feature information that can be extracted. To address these issues, this paper improves the backbone network and feature extraction network of the YOLOv5 model in terms of two aspects: reducing the model weight and improving detection accuracy. The improved model is called YOLOv5-GBC, and its network structure is shown in Figure 1.

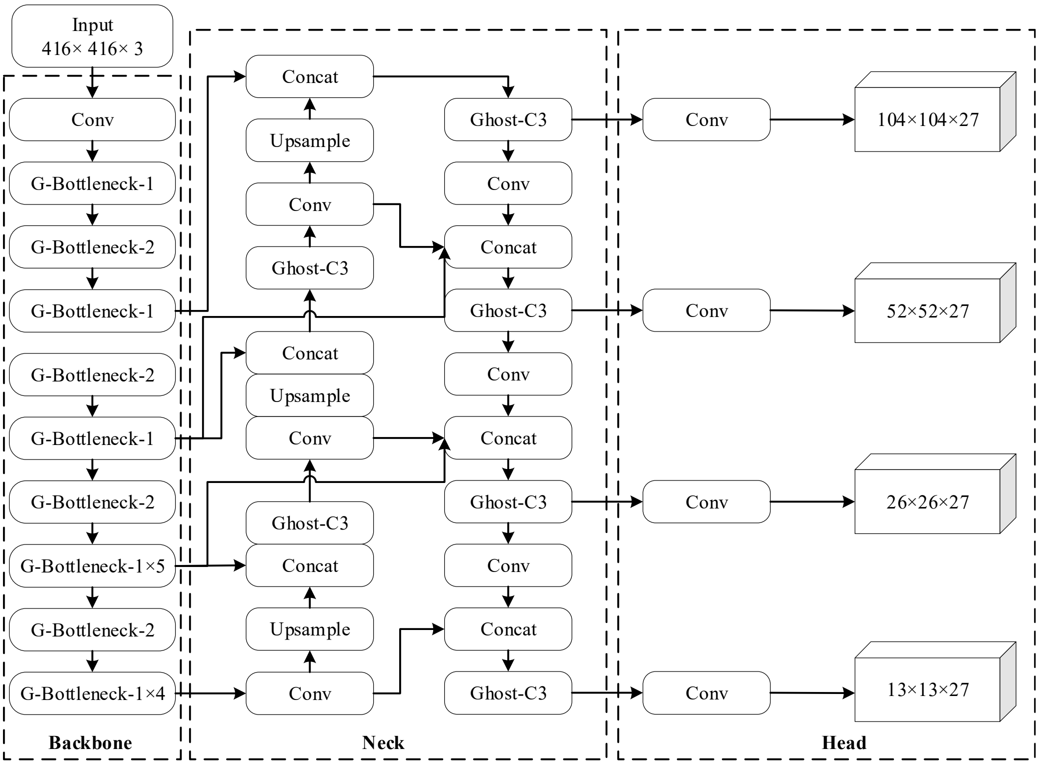


Figure 1. Network structure diagram of YOLOv5-GBC.

YOLOv5-GBC mainly consists of four parts: the input terminal, backbone network, neck network, and prediction head.

(1) Input terminal: In order to expand the number of datasets and enrich the background of the detected objects, traditional data augmentation techniques such as random scaling, flipping, and color jitter, as well as Mosaic data augmentation techniques, are used to process the input images, improving the model's robustness.

(2) Backbone network: The original YOLOv5 backbone network consists of four different-sized bottleneck modules, but traditional convolutional approaches have a large computational load. In order to meet the demand for lightweight mobile deployment, the lightweight network, GhostNet, which is suitable for mobile platforms, is used instead of the original backbone network, as shown in Figure 1. The improved backbone network can maintain high detection accuracy while greatly reducing the network parameter count and floating-point computation load. Secondly, to improve the backbone network's feature extraction capabilities, a coordinate attention mechanism is introduced in the feature fusion network to enhance the model's sensitivity to different feature channels and reduce the influence of interference information.

(3) Neck network: The original YOLOv5 neck network mainly consists of a Fast Spatial Pyramid Pooling (SPPF) module and a Path-Aggregation Network (PANet) [17]. PANet improves the fusion effect between deep semantic information and shallow spatial information based on the Feature Pyramid Networks (FPNs) [18]. However, this approach can require a large number of parameters and computational loads. To further improve the model's feature fusion efficiency, this paper improves the original neck network based on the Weighted Bidirectional Feature Pyramid Network (BiFPN) using weighted feature fusion and same-scale feature residual connections. On the other hand, to improve detection accuracy for tiny defects, shallow features containing rich spatial information are fused into the neck network.

(4) Prediction head: The prediction head is mainly used to process the multi-scale feature maps generated by the neck network, generating position, confidence, and category information for the predicted bounding boxes. In response to the improvement of the neck network, a prediction head dedicated to detecting tiny targets is added.

2.1. Lightweight Backbone Network

Traditional convolution operations generate a large number of redundant feature maps, which often contain many similar parts [19]. Although redundant feature maps can effectively improve the detection accuracy of the model, a large number of convolution operations will also increase the calculation load, which is not conducive to lightweight deployment of the model.

To address this issue, this paper proposes a lightweight network called GhostNet [20] based on the Ghost module to improve the original backbone network. Compared with other lightweight networks such as MobileNet [21] and ShuffleNet [22], GhostNet has better detection performance and can significantly reduce the computational cost required for ordinary convolutions while maintaining detection accuracy stability. The main idea is to obtain redundant feature layer information through low-cost linear operations, thereby generating more feature information with fewer parameters. The convolution process of the Ghost module is shown in Figure 2, and mainly consists of three steps. The first step is to generate some feature maps through traditional convolution to avoid redundancy. The second step is to generate another part of redundant feature maps through simple linear transformations of the generated feature maps. The third step is to superimpose these two parts of feature maps to achieve the effect of simulating ordinary convolution.

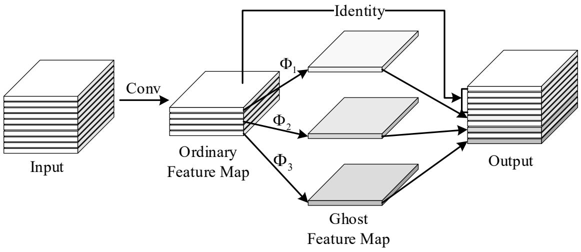


Figure 2. The Ghost convolution process.

Taking $k \times k$ ordinary convolution as an example, when the number of input feature channels is C and the output feature size is $H \times W \times N$, the computational cost is:

$$C \times k \times k \times N \times H \times W \tag{1}$$

Assuming the number of output feature channels in the first step of the Ghost module is m, $s = N/m$ there are $(s-1) \times m$ linear transformations in the second step, and the size of the convolution kernel is the same $k \times k$. The computational cost is:

$$C \times k \times k \times m \times H \times W + (s-1) \times k \times k \times m \times H \times W \tag{2}$$

As can be seen, the ratio of computational cost of the model is:

$$r_s = \frac{C \times k \times k \times N \times H \times W}{(C+s-1) \times k \times k \times m \times H \times W} = \frac{C \times s}{C+s-1} \approx s \tag{3}$$

The ratio of model size is:

$$r_c = \frac{C \times k \times k \times N}{C \times k \times k \times m + (s-1) \times k \times k \times m} = \frac{C \times s}{C+s-1} \approx s \tag{4}$$

Compared to regular convolution, both the parameter and computational cost compression ratios of the Ghost module are S, which can effectively reduce the size of the model and improve detection speed. In this paper, s is set to 2.

The bottleneck structure composed of Ghost modules consists of two cases: The G-Bottleneck-1 structure has a stride of 1, consisting of two Ghost modules. The first module expands the channel number of the convolution, the second module reduces the channel number, and the input and output are added through a residual edge, as shown in Figure 3a. The G-Bottleneck-2 structure contains two Ghost modules and a depth-wise separable convolution module, in which the stride of the depth-wise separable convolution is 2, thus compressing the width and height of the feature layer, as shown in Figure 3b. Using G-Bottleneck-1 and G-Bottleneck-2 to construct the lightweight feature extraction network GhostNet, and replacing the original backbone network in YOLOv5, can effectively reduce the volume and computing power of the model, as shown in the network structure in Figure 1.

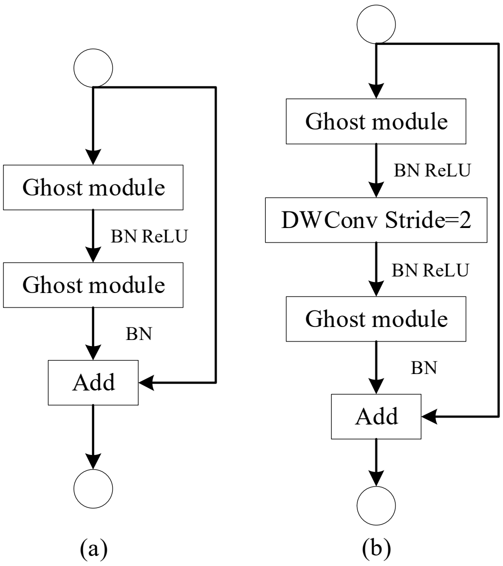


Figure 3. G-Bottleneck network structure. (**a**) G-Bottleneck-1; (**b**) G-Bottleneck-2.

2.2. Improving the Feature Fusion Network

In the process of feature extraction, shallow features usually contain detailed spatial information, deep features contain rich semantic information, and single-scale features often have limited representation capabilities. Therefore, effective fusion of features of different scales is crucial for object detection tasks.

Traditional FPN networks only have one top-down information fusion path. In order to improve its one-way structure, PANet adds a bottom-up path based on FPN, which improves the fusion effect between high-level semantic information and low-level spatial information. However, this method also has a larger computational cost and does not consider the contribution of different input features. Therefore, based on the idea of weighted feature fusion and feature residual connection in BiFPN, this paper constructs a new feature fusion network called WR-PANet, as shown in Figure 4.

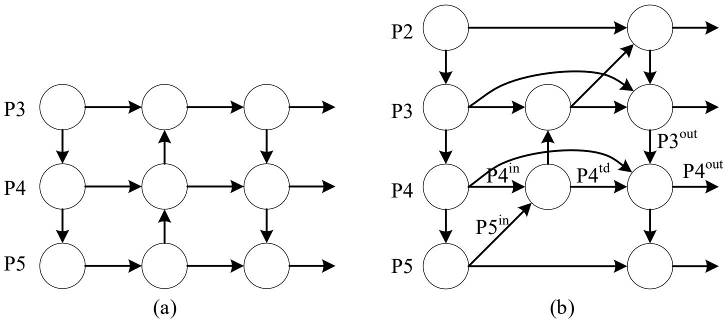


Figure 4. Feature fusion network. (**a**) PANet; (**b**) WR-PANet.

As shown in Figure 4, compared with the PANet network, in order to improve the efficiency of feature fusion, BiFPN ignores nodes with only one input edge. Secondly, BiFPN adds a residual connection between feature layers of the same scale, which can fuse more features and improve the positioning ability of defect targets without increasing computational cost.

Since different input features have different scales and different contribution levels compared to the nodes in feature fusion, BiFPN adopts a weighted feature fusion method and adds additional weight parameters for different input features. Weighted feature fusion uses a fast normalization calculation method, as shown in Formula (5).

$$O = \sum_i \frac{\omega_i \times I_i}{\varepsilon + \sum_j \omega_j} \tag{5}$$

In the equation, ω_i is the weight of different input features, by network training, I_i is the input feature, and ε is the initial learning rate.

$$\begin{aligned} P4^{td} &= Conv\left(\frac{\omega_1 \times P4^{in} + \omega_2 \times resize(P5^{in})}{\omega_1 + \omega_2 + \varepsilon}\right) \\ P4^{out} &= Conv\left(\frac{\omega_1' \times P4^{in} + \omega_2' \times P4^{td} + \omega_3' \times resize(P3^{out})}{\omega_1' + \omega_2' + \omega_3' + \varepsilon}\right) \end{aligned} \tag{6}$$

The process of feature fusion is shown as Formula 6. In the equation, $P4^{td}$ and $P4^{out}$ are the intermediate feature and output characteristic of the fourth layer, $P5^{in}$ is the input feature of the fifth layer, $P3^{out}$ is the output feature of the third layer, resize is the up sampling and down sampling operation, and $Conv$ represents the convolution operation.

On the other hand, because defects such as cracks are usually small, as the network deepens, the backbone network inevitably loses more detailed information during the down sampling process, which leads to poor performance in detecting small targets. Unlike deep networks that contain rich semantic information, shallow networks contain more spatial position information and are more helpful in determining the accurate location of targets. Therefore, this paper transfers the features extracted from a shallow layer to the neck network for fusion, and adds a prediction head for detecting small targets to improve the detection ability of small defects.

2.3. Coordinate Attention Mechanism

An attention mechanism can assign weights according to the importance of different features, assigning higher weights to key features such as target textures to increase sensitivity, and lower weights to interference information such as noise to reduce their impact on detection performance. Due to its good performance and plug-and-play characteristics, it has achieved good results in various computer vision tasks. Currently, mainstream attention mechanisms are mainly divided into two categories; one is the channel attention mechanism represented by Squeeze-and-Excitation (SE) [23] and Efficient Channel Attention (ECA) [24], and the other is the hybrid attention mechanism represented by the

Convolutional Block Attention Module (CBAM) [25]. The channel attention mechanism only considers the relationship between feature channels and ignores the importance of spatial information on target detection. CBAM considers both channel and spatial information, but due to the limitations of its structure, it only considers local area information and cannot effectively use the overall spatial information. Moreover, these methods have a large computational burden and are not suitable for lightweight network models.

In sewer defect detection tasks, on the one hand, due to the low image resolution and poor lighting in the pipeline, a large amount of noise and interference information can easily be generated; on the other hand, cracks and other small defects occupy fewer pixels, which can be easily affected by interference information and result in information loss during feature extraction. To address the above issues, this paper introduces the coordinate attention (CA) mechanism, which can simultaneously focus on channel information and positional information in the Ghost module, enhancing the model's sensitivity to key features for defect detection and further improving the detection accuracy of defects. The CA attention mechanism can also improve the detection accuracy of the model with minimal overhead, making it more suitable for model deployment on mobile devices. The overall structure of the CA mechanism is shown in Figure 5.

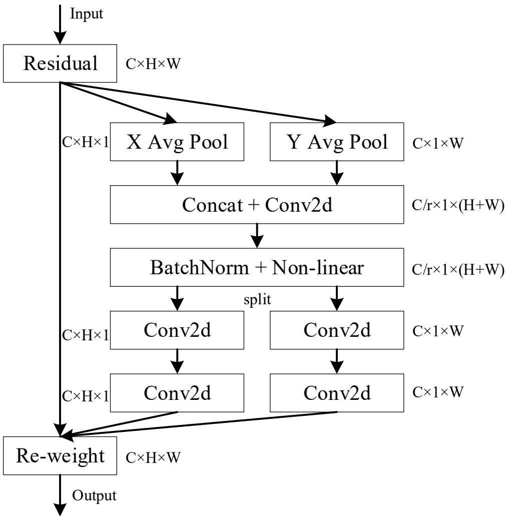


Figure 5. Coordinate attention mechanism.

The CA mechanism mainly consists of two steps: information embedding and attention generation. Information embedding: Perform horizontal and vertical average pooling operations on the input features separately, outputting a pair of feature maps with orientation perception capabilities, with sizes of $C \times H \times 1$ and $C \times 1 \times W$. Attention generation: Firstly, a series of concatenation operations are performed on the two feature maps obtained above and ReLU nonlinear activations are performed on the two obtained feature maps to generate the intermediate feature f, with a size of $C/r \times 1 \times (W + H)$. Then, the intermediate feature f is decomposed into horizontal and vertical tensors along the spatial dimension, respectively, and then up sampled by convolution operations and activated by the Sigmoid function to obtain the attention weights in the horizontal and vertical directions. Finally, the attention weights are multiplied by the input features to complete the application of attention weights.

Since the shallow features are fused in the neck network and an additional detection head is added, the model becomes more complex, which is not conducive to lightweight deployment. Therefore, by introducing the CA mechanism into the G-Bottleneck-1 structure and replacing the bottleneck structure in the original C3 module, the lightweight processing of the neck network is achieved. The improved C3 module is called Ghost-C3, and its network structure is shown in Figure 6.

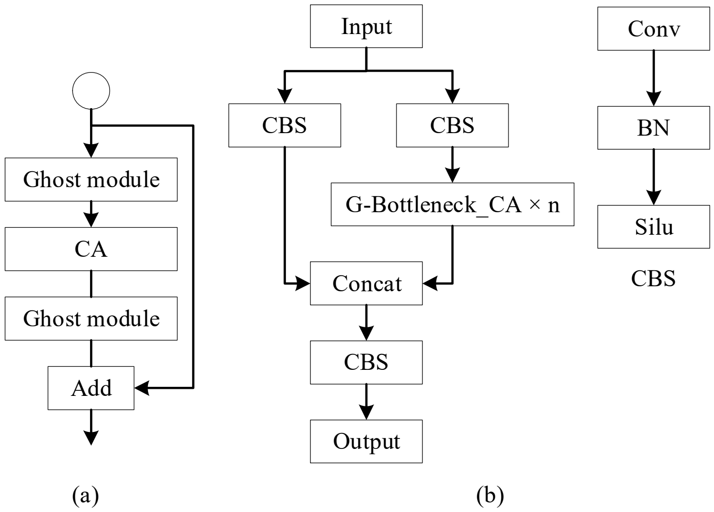


Figure 6. G-Bottleneck module with CA mechanism and Ghost-C3 module. (**a**) G-Bottleneck-CA; (**b**) Ghost-C3.

3. Experimental Results and Analysis

3.1. Data Preparation

The data used in this study came from the Sewer-ML [26] sewer defect classification dataset published by a Danish laboratory. The dataset contains different defect categories in various pipeline backgrounds, all of which are classified by professional technicians, making it an ideal source of data for studying pipeline defect detection. Since Sewer-ML contains numerous defect categories and does not provide defect location information, it cannot be directly applied to object detection tasks. Therefore, this paper selected 2700 images from Sewer-ML for defect location annotation, including four common drainage pipeline defects in the southern region of China: cracks, stagger, deposition, and root [27]. Figure 7 shows some images of pipeline defects. The position marked by the rectangular box in the figure is the position of the defect. It can be seen that there are multiple defects coexisting in the actual pipeline, which increases the difficulty of detection to a certain extent. Table 1 shows the distribution of each defect label in the dataset of this study.

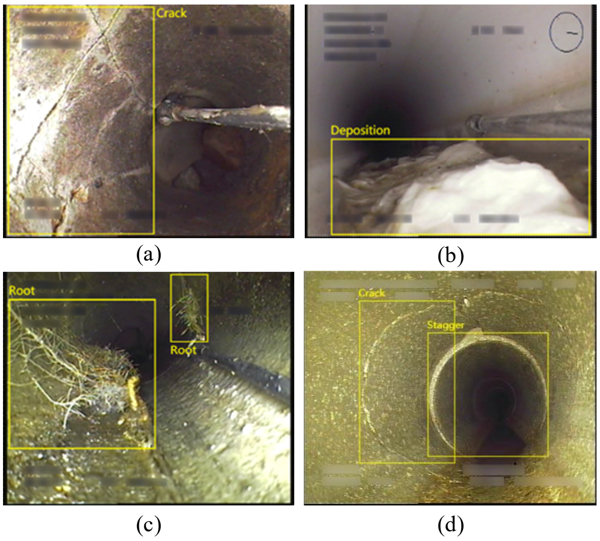


Figure 7. Example images of sewer defects. (**a**) crack; (**b**) deposition; (**c**) root; and (**d**) stagger, crack.

Table 1. Distribution of defect labels.

Defect	Number
Crack	749
Deposition	780
Stagger	778
Root	748
Total	3055

When the number of images is small, it may lead to low model accuracy and there is a risk of overfitting. In order to avoid these problems, this paper not only uses the Mosaic data augmentation method, but also uses traditional data augmentation techniques to increase the size of the dataset. Data augmentation techniques are very important for datasets with few images, as they not only increase the size of the dataset but also increase the diversity of input images, making the designed model more adaptable to complex backgrounds and improving its robustness. Traditional data augmentation methods usually include color transformation and geometric transformation methods, where color transformation includes methods such as contrast, hue, and saturation transformation, and geometric transformation includes methods such as random scaling, flipping, and cropping.

In this paper, four methods, namely, random scaling, random horizontal flipping, random vertical flipping, and color jitter (randomly adjusting the brightness, contrast, saturation, and hue of the image), were used to enhance each image, and the effects of each data augmentation technique are shown in Figure 8. The enhanced images were all scaled to 416 × 416 pixels for model training.

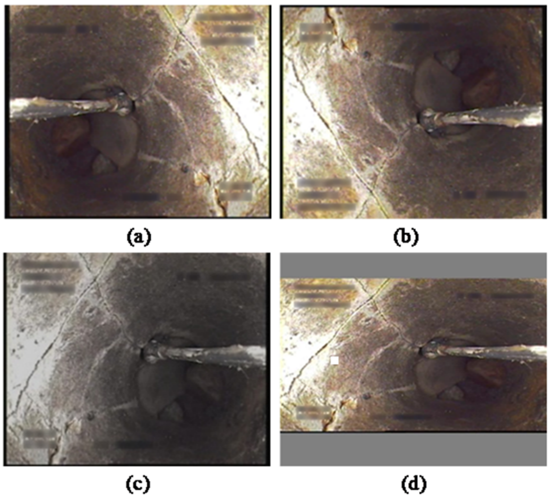


Figure 8. Data augmentation methods. (**a**) horizontal flipping; (**b**) vertical flipping; (**c**) color jitter; and (**d**) random scaling.

3.2. Experimental Environment and Configuration

The training environment of the model is shown in Table 2. In the test environment of this study, the CPU was an Intel i5-11300H (Intel Corporation, Santa Clara, CA, USA), the memory was 16G, and the operating system was Windows, which was used to simulate the terminal control equipment in the CCTV detection system. During the model training and testing process, the training set, validation set, and test set were divided in a ratio of 8:1:1. The initialization parameters during training were as follows: the input size of the image was set to 416 × 416, the total number of iterations (epochs) was set to 300, the batch size was set to 8. The Adam optimizer was used to optimize the loss function and the cosine annealing algorithm was used to adjust the learning rate. The pre-trained weights of the model on the VOC dataset were loaded through transfer learning to accelerate the convergence speed of the network.

Table 2. Experimental environment configuration.

Environment	Parameters
Operating System	Ubuntu
Deep Learning Framework	PyTorch 1.7.0
CPU	Intel(R) Xeon(R) Platinum 8255C
GPU	RTX 3080

The loss curve during the training and validation process is shown in Figure 9. It can be seen that the model converged after 300 epochs of training, and there is no overfitting or underfitting.

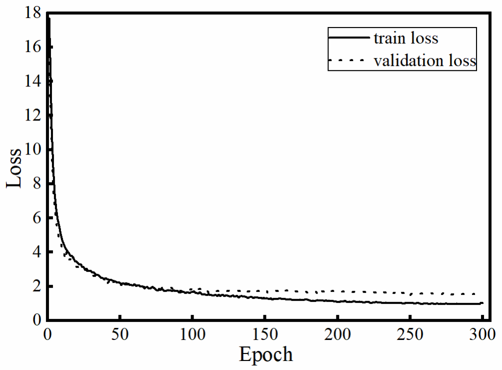


Figure 9. Loss curves during training and validation.

3.3. Evaluation Metrics

This study used the recall rate (Recall, R), precision (Precision, P), F1 score, and average precision mean (mAP) to evaluate the defect detection and classification performance of different models. In addition, the model's parameter size, model volume, floating-point operation count (FLOPs), and inference speed (FPS) were used to evaluate the size and detection speed of the model.

3.4. Analysis of Ablation Experiment

In order to verify the effect of the improvement strategy on the model performance in this study, each module was evaluated under the same conditions. The results of the ablation experiments are shown in Table 3; the '√' sign indicates the addition of the corresponding improvement strategy.

Table 3. Ablation experiment results.

Model	GhosNet	WR-PANet	CA	mAP@0.5 /%	Recall /%	F1	Model Parameters /10^6	Model Volume /MB	FLOPs /10^9
YOLOv5	-	-	-	86.33	80.92	0.80	46.40	177.95	48.42
YOLOv5-G	√	-	-	86.06	79.82	0.81	22.32	85.15	17.45
YOLOv5-GB	√	√	-	88.17	79.78	0.82	23.30	90.01	26.47
YOLOv5-GBC	√	√	√	87.21	82.43	0.84	12.06	46.01	12.21

(1) Influence of lightweight backbone network on model performance.

As shown in the above table, YOLOv5-G represents using GhostNet as the backbone network. The first row of the table represents using the original YOLOv5 model for defect detection, with an mAP of 86.33%, an average recall rate of 80.92%, and an average F1 value of 0.80. It can be seen that compared with YOLOv5, the mAP of the YOLOv5-G model is reduced by 0.27%, and the average recall rate is reduced by 1.10%. On the other hand, the model parameters, volume, and floating-point operation count of YOLOv5-G are reduced by 51.90%, 52.15%, and 63.96%, respectively. This indicates that the lightweight model YOLOv5-G significantly reduces the volume and floating-point operation count of the model while keeping the detection accuracy unchanged, proving the superiority of the improved backbone network.

Table 4 shows the AP, recall rate, and detection speed of each model for different defect classes on CPU and GPU. It can be seen that the detection accuracy of YOLOv5-G for cracks decreases by 6.99%, which may be due to the Ghost module using simple linear operations to generate redundant feature maps. For small targets with unclear features, the Ghost module's feature extraction capabilities for these targets are reduced, leading to a decrease in detection accuracy. The FPS index shows that the detection speed of YOLOv5-G increased by 87.27%, combined with the significant reduction in the volume and floating-point operation count of the YOLOv5-G model, indicating the superior performance of the improved backbone network.

Table 4. AP, Recall and FPS of different models for each defect.

Model	**AP50/%**				**Recall/%**				**FPS /(Frames/s)**
	Crack	**Deposition**	**Root**	**Stagger**	**Crack**	**Deposition**	**Root**	**Stagger**	
YOLOv5	74.26	89.21	90.68	91.17	61.82	80.95	90.24	90.67	5.5
YOLOv5-G	67.21	93.72	89.27	94.05	58.90	87.95	79.27	93.15	10.3
YOLOv5-GB	81.44	93.01	85.05	93.17	67.12	85.54	71.95	94.52	8.4
YOLOv5-GBC	82.20	94.85	81.14	90.64	68.49	93.98	76.83	90.41	9.0

(2) The influence of the improved feature fusion network on the model performance.

YOLOv5-GB indicates the use of an improved feature fusion network, WR-PANet, based on the YOLOv5-G model. From the experimental data in Table 3, it can be seen that the improved model has better detection accuracy. Among them, the mAP increased by 2.11%, and the recall rate and F1 value changed slightly. On the other hand, due to the more complex neck network after improvement, the model volume and floating-point calculation power increased by 5.8% and 51.69%, respectively, which were still less than those of the YOLOv5 model. From the experimental data in Table 4, it can be seen that due to the fusion of shallow features with richer spatial information in the neck network, the improved model greatly increased the detection accuracy and recall rate of cracks, by 8.22% and 14.23%, respectively, and also had good detection effects on the other three types of defects. On the other hand, due to the more complex network, the detection speed of the model on the CPU was reduced by 18.44%. Overall, although some detection speed was sacrificed, the improved neck network significantly improved the model's detection accuracy, especially for defects such as cracks.

(3) The impact of the Ghost module with the added CA mechanism on model performance.

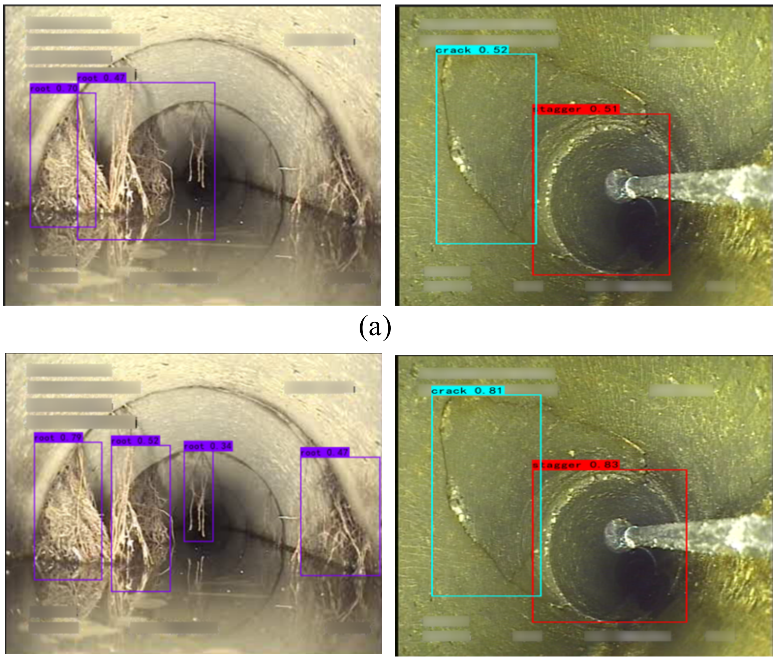
YOLOv5-GBC indicates the lightweight processing of the neck network with the introduction of the attention mechanism Ghost module based on the YOLOv5-GB model. It can be seen that although the mAP has decreased to some extent, the model volume and FLOPs have been significantly reduced by 48.88% and 53.87%, respectively. As the attention mechanism is added, the model can pay more attention to the feature information that is useful for defect detection during feature fusion, and suppress the influence of noise and other interfering information; the average recall rate and F1 value of the model are increased by 3.65% and 2%, respectively. From Table 4, it can be seen that due to the addition of the attention mechanism, FPS on the GPU has been slightly reduced, but the introduction of the Ghost module has increased the detection speed of the model on the CPU. Although the Ghost module with the CA mechanism sacrifices some detection accuracy, it greatly reduces the model volume and calculation power, and ultimately improves the model's detection accuracy while significantly reducing the required calculation power, making it more suitable for pipeline defect detection tasks on small mobile devices.

3.5. Comparison Analysis of the Model before and after Improvement

The data in Table 5 can be obtained from the above ablation experiments. It can be seen that compared with the original YOLOv5, the improved model, YOLOv5-GBC, has increased mAP and recall rate by 0.88% and 1.51%, respectively, and precision has increased by 5.31%. The model volume and FLOPs have been reduced by 74.15% and 74.78%, respectively. In the test environment, the detection speed of YOLOv5-GBC has increased by 63.64%. Overall, the improved YOLOv5 algorithm further improves detection performance while achieving a balance between lightness and accuracy, and solves problems such as complex existing model structures that cannot be used on small mobile devices. In order to better verify the detection performance of the above models on the pipeline defect dataset, YOLOv5-GBC is used to detect pipeline defect images under different backgrounds, and the results are shown in Figure 10, where Figure 10a represents the detection result of YOLOv5 and Figure 10b represents the detection result of YOLOv5-GBC. The red boxes are stagger, the purple boxes are root, the blue boxes are crack, and the green boxes are deposition. From the two sets of contrasted figures, it can be seen that in the first group, the YOLOv5-GBC model is more accurate in locating tree roots, and has a higher recognition rate, while the original YOLOv5 model has more missed detections. In the second group, the two models have similar detection effects, but the YOLOv5-GBC model has a prediction confidence that is about 30% higher for cracks and displacements, and the predicted boxes are more accurate. It can be seen that both models have the ability to detect various types of defects in complex environments, but the improved model has better detection performance and can detect more defects under the same conditions.

Table 5. Comparison of detection results before and after model improvement.

Model	mAP@0.5/%	Recall/%	Precision/%	F1	Model Parameters /10^6	Model Volume /MB	FLOPs /109	FPS /(Frames/s)
YOLOv5	86.33	80.92	80.07	0.80	46.40	177.95	48.42	5.5
YOLOv5-GBC	87.21	82.43	85.38	0.84	12.06	46.01	12.21	9.0



(a)

(b)

Figure 10. Comparison of test results before and after model improvement. (**a**) YOLOv5 detection results; (**b**) YOLOv5-GBC detection results.

3.6. Comparison Analysis with Other Models

To further verify the effectiveness of the improved model, this paper compares YOLOv5-GBC with mainstream object detection models under the same experimental conditions. Other detection models include single-stage detection models SSD, YOLOv3, YOLOv4, and YOLOv7, the dual-stage detection model Faster-RCNN, the anchor-free detection model YOLOX, and YOLOv5-M, in which the original YOLOv5 backbone network is replaced with the MobileNet network. The experimental results are shown in Table 6.

Table 6. Comparison of detection results of different object detection models.

Model	mAP@0.5 /%	Recall/%	Precision/%	F1	Model Parameters /10^6	Model Volume /MB	FLOPs/10^9
SSD	82.38	80.79	75.87	0.78	24.01	91.6	115.97
YOLOv3	84.57	80.31	85.18	0.83	61.54	234.76	65.62
YOLOv4	86.76	85.17	86.32	0.86	63.95	243.96	59.98
YOLOX	89.89	89.78	86.04	0.87	54.15	206.57	65.78
YOLOv7	88.34	81.41	90.9	85.5	37.21	141.95	44.42
Faster RCNN	79.79	88.07	49.02	0.63	136.75	521.66	252.66
YOLOv5-M	85.61	81.29	84.08	0.82	22.63	86.34	18.05
YOLOv5-GBC	87.21	82.43	85.38	0.84	12.06	46.01	12.21

As shown in Table 6, the YOLOX model achieved the highest mAP, but its model volume and FLOPs are higher, making it unsuitable for mobile devices. The two-stage network Faster RCNN has the highest recall rate, but its detection accuracy is poor due to its lower precision, and its model volume and FLOPs are the largest, making it the slowest in terms of detection speed and unsuitable for detection tasks on mobile platforms. The single-stage network YOLO series models have good detection accuracy, but their volume and computation are large, and they cannot meet the lightweight deployment requirements on mobile devices. Compared with other models, the improved model, YOLOv5-GBC, proposed in this paper has significant advantages in terms of model volume and FLOPs, and its mAP is only lower than that of the optimal YOLOX model, while recall rate and F1 value are comparable to other models; thus, it can meet the high precision requirements of pipeline defect detection tasks. Overall, the model proposed in this paper has lower model volume and computation while maintaining high detection accuracy, making it more suitable for small mobile devices with limited computing resources.

In order to verify the defect detection ability of different models, we selected five representative models to conduct experiments under the same conditions, and the detection results are shown in Figure 11. It can be seen that our improved model has the ability to accurately detect different types of defects in complex environments and is comparable to the detection results of the YOLOX. Compared with other detection models, it has better detection results.

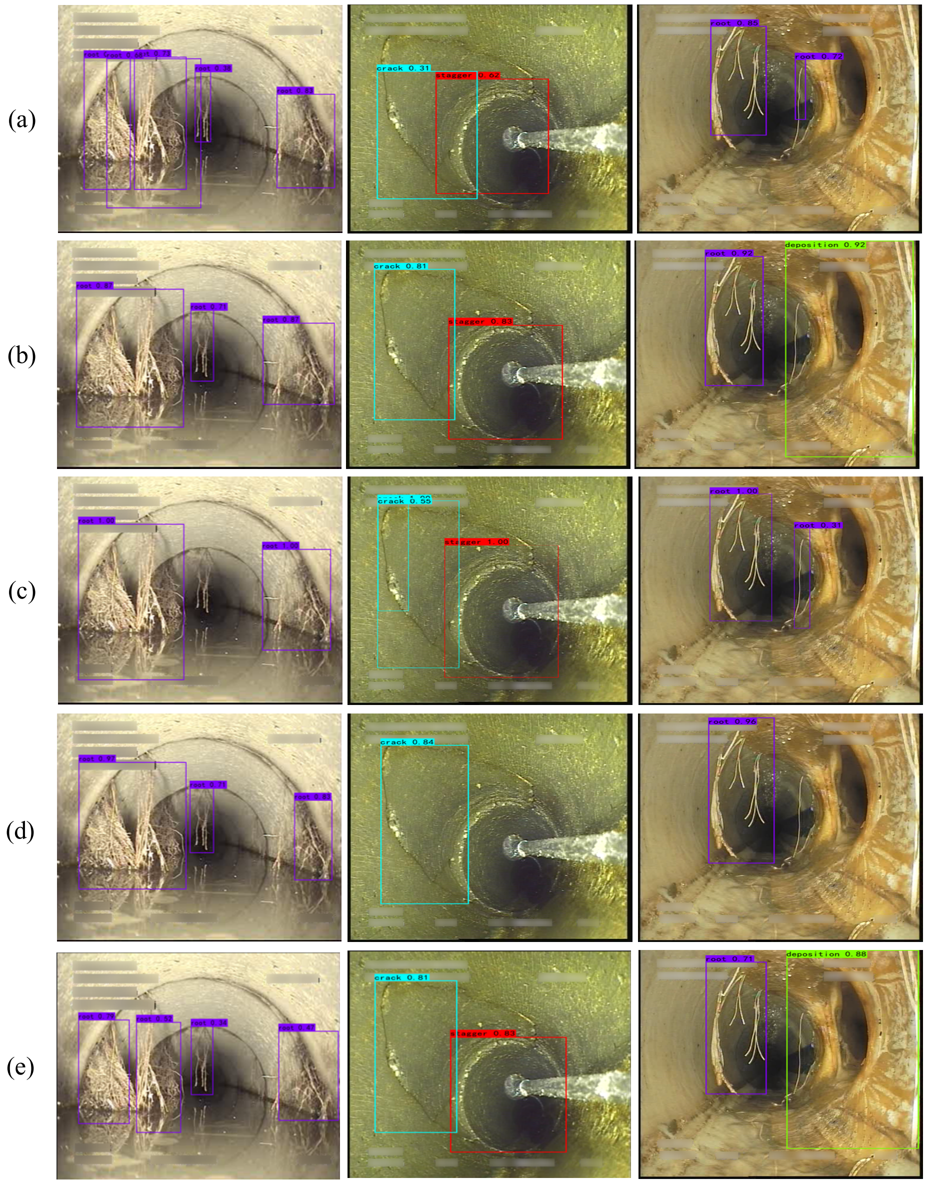


Figure 11. Some detection results of sewer defect images using different models: (**a**–**d**) show the detection results by YOLOv3, YOLOX, Faster R-CNN, and YOLOv5-M, respectively; (**e**) shows the detection results of our model.

4. Conclusions

Rapid and comprehensive detection of drainage sewer defects is crucial for maintaining urban safety. This paper proposes a lightweight pipeline defect detection model based on the improvement of YOLOv5. By improving the backbone network and feature fusion network and introducing an attention mechanism, while improving the detection accuracy of the model, the volume and computational complexity of the model are significantly reduced. Experimental results show that the improved model has the ability to accurately classify and detect multiple sewer defects at the same time. The model volume and FLOPs are reduced by 74.15% and 74.78%, respectively, and the mAP and F1 values are improved by 0.88% and 1.51%, achieving a balance between light weight and accuracy, and better meeting the deployment requirements on small mobile devices. Due to the fact that drainage sewers are easily subject to various factors during long-term use, other types of defects may also appear in the sewers, which seriously affect the normal operation of the sewer system. In the future, more types of defect images will be collected to meet the model's detection requirements for other defects.

Author Contributions: Methodology, X.Z. and X.L.; investigation, L.T.; writing—original draft preparation, X.Z. and J.Z.; writing—review and editing, X.L. and S.W.; funding acquisition, X.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the China University Industry-University-Research Innovation Fund (2021FNB02001).

Institutional Review Board Statement: This study did not require ethical approval.

Informed Consent Statement: This study did not involve humans.

Data Availability Statement: Not applicable.

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

References

1. Wang, Q.; Yao, J.; Tan, W.L.; Pan, H.H. Research on Defect Detection of Drainage Pipeline Based on Faster R-CNN. *Softw. Guide* **2019**, *18*, 40–44.
2. Liu, X.Y.; Ye, S.Z.; Lü, B.; Yan, Z. Information Solution for Intelligent Detection of Drainage Pipe Network Defects. *China Water Wastewater* **2021**, *37*, 32–36.
3. Wang, H.H.; Xie, H.L.; Gao, Y.; Liu, J.X.; Song, X.F. Evaluation method of municipal sewer health status based on YOLO v5. *Water Wastewater Eng.* **2022**, *58*, 130–136.
4. Shankar, R.S.; Srinivas, L.V.; Neelima, P.; Mahesh, G. A Framework to Enhance Object Detection Performance by Using YOLO Algorithm. In Proceedings of the 2022 International Conference on Sustainable Computing and Data Communication Systems (ICSCDS), IEEE, Erode, India, 7–9 April 2022; pp. 1591–1600.
5. Devareddi, R.B.; Shankar, R.S.; Murthy, K.; Raminaidu, C. Image segmentation based on scanned document and hand script counterfeit detection using neural network. *AIP Conf. Proc.* **2022**, *2576*, 050001.
6. Li, Y.; Wang, H.; Dang, L.M.; Song, H.K.; Moon, H. Vision-based defect inspection and condition assessment for sewer pipes: A comprehensive survey. *Sensors* **2022**, *22*, 2722. [CrossRef] [PubMed]
7. Zhou, Q.; Situ, Z.; Teng, S.; Chen, W.; Chen, G.; Su, J. Comparison of classic object-detection techniques for automated sewer defect detection. *J. Hydroinform.* **2022**, *24*, 406–419. [CrossRef]
8. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. Ssd: Single Shot Multibox Detector. In Proceedings of the Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016; Part I 14. Springer: Berlin/Heidelberg, Germany, 2016; pp. 21–37.
9. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
10. 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]
11. Yin, X.; Chen, Y.; Bouferguene, A.; Zaman, H.; Al-Hussein, M.; Kurach, L. A deep learning-based framework for an automated defect detection system for sewer pipes. *Autom. Constr.* **2020**, *109*, 102967. [CrossRef]
12. Cheng, J.C.P.; Wang, M. Automated detection of sewer pipe defects in closed-circuit television images using deep learning techniques. *Autom. Constr.* **2018**, *95*, 155–171. [CrossRef]
13. Li, D.; Xie, Q.; Yu, Z.; Wu, Q.; Zhou, J.; Wang, J. Sewer pipe defect detection via deep learning with local and global feature fusion. *Autom. Constr.* **2021**, *129*, 103823. [CrossRef]
14. Huang, D.; Liu, X.; Jiang, S.; Wang, H.; Wang, J.; Zhang, Y. Current state and future perspectives of sewer networks in urban China. *Front. Environ. Sci. Eng.* **2018**, *12*, 2. [CrossRef]
15. Tan, M.; Pang, R.; Le, Q.V. Efficientdet: Scalable and Efficient Object Detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 10781–10790.
16. Hou, Q.; Zhou, D.; Feng, J. Coordinate Attention for Efficient Mobile Network Design. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 13713–13722.
17. Liu, S.; Qi, L.; Qin, H.; Shi, J.; Jia, J. Path Aggregation Network for Instance Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 8759–8768.
18. Lin, T.-Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature Pyramid Networks for Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Dhaka, Bangladesh, 13–14 February 2017; pp. 2117–2125.
19. Wang, K.; Xu, M.; Sun, X.; Xu, Q.; Tan, S.-B. Insulator Self-explosion Defect Detection Method Based on Improved YOLOv3. *J. Chin. Comput. Syst.* **2022**, *43*, 2564–2569.

20. Han, K.; Wang, Y.; Tian, Q.; Guo, J.; Xu, C.; Xu, C. Ghostnet: More Features from Cheap Operations. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 1580–1589.
21. Howard, A.; Sandler, M.; Chu, G.; Chen, L.-C.; Chen, B.; Tan, M.; Wang, W.; Zhu, Y.; Pang, R.; Vasudevan, V.; et al. Searching for Mobilenetv3. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 1314–1324.
22. Ma, N.; Zhang, X.; Zheng, H.T.; Sun, J. Shufflenet v2: Practical Guidelines for Efficient cnn Architecture Design. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 116–131.
23. Hu, J.; Shen, L.; Sun, G. Squeeze-and-Excitation Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7132–7141.
24. Wang, Q.; Wu, B.; Zhu, P.; Li, P.; Zuo, W.; Hu, Q. ECA-Net: Efficient Channel Attention for Deep Convolutional Neural Networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 11534–11542.
25. Woo, S.; Park, J.; Lee, J.Y.; Kweon, I.S. Cbam: Convolutional Block Attention Module. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 3–19.
26. Haurum, J.B.; Moeslund, T.B. Sewer-ML: A Multi-Label Sewer Defect Classification Dataset and Benchmark. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 13456–13467.
27. Lin, M.B. Health inspection and analysis of sewer system in an area of Fuzhou City. *China Water Wastewater* **2014**, *30*, 96–98.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.