

Article

PDS-YOLO: A Real-Time Detection Algorithm for Pipeline Defect Detection

Ke Zhang ¹, Longxiao Qin ^{1,*}  and Liming Zhu ^{2,*}

¹ School of Mechanical Engineering, Shanghai Institute of Technology, Shanghai 201418, China; zhangk2010@sit.edu.cn

² SGIDI Engineering Consulting (Group) Co., Ltd., Shanghai 200093, China

* Correspondence: 226092117@mail.sit.edu.cn (L.Q.); zhuliming@sgidi.com (L.Z.)

Abstract: Regular inspection of urban drainage pipes can effectively maintain the reliable operation of the drainage system and the production safety of residents. Aiming at the shortcomings of the CCTV inspection method used in the drainage pipe defect detection task, a PDS-YOLO algorithm that can be deployed in the pipe defect detection system is proposed to overcome the problems of inefficiency of manual inspection and the possibility of errors and omissions. First, the C2f-PCN module was introduced to decrease the model sophistication and decrease the model weight file size. Second, to enhance the model's capability in detecting pipe defect edges, we incorporate the SPDSC structure within the neck network. Introducing a hybrid local channel MLCA attention mechanism and Wise-IoU loss function based on a dynamic focusing mechanism, the model improves the precision of segmentation without adding extra computational cost, and enhances the extraction and expression of pipeline defect features in the model. The experimental outcomes indicate that the mAP, F1-score, precision, and recall of the PDS-YOLO algorithm are improved by 3.4%, 4%, 4.8%, and 4.0%, respectively, compared to the original algorithm. Additionally, the model achieves a reduction in both the model's parameter and GFLOPs by 8.6% and 12.3%, respectively. It saves computational resources while improving the detection accuracy, and provides a more lightweight model for the defect detection system with tight computing power. Finally, the PDS-YOLOv8n model is deployed to the NVIDIA Jetson Nano, the central console of the mobile embedded system, and the weight files are optimized using TensorRT. The test results show that the velocity of the model's inference capabilities in the embedded device is improved from 5.4 FPS to 19.3 FPS, which can basically satisfy the requirements of real-time pipeline defect detection assignments in mobile scenarios.



Academic Editor: Honarvar Shakibaei Asli

Received: 6 December 2024

Revised: 29 December 2024

Accepted: 4 January 2025

Published: 6 January 2025

Citation: Zhang, K.; Qin, L.; Zhu, L. PDS-YOLO: A Real-Time Detection Algorithm for Pipeline Defect Detection. *Electronics* **2025**, *14*, 208. <https://doi.org/10.3390/electronics14010208>

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: defect detection; lightweight model; model deployment; mobile embedded device; YOLOv8

1. Introduction

As society continues to progress, the economy is booming, the scale of the city continues to expand, and the complexity of urban underground drainage pipelines continues to increase, resulting in a substantial increase in the difficulty of conducting pipeline condition assessment, and the maintenance of underground drainage pipelines is seriously affected [1]. Pipelines are affected by factors such as the increase in service life and changes in the external environment, and various types of defects may occur, leading to breakage, scaling, obstacle blockage, and other safety hazards, resulting in a reduction in the pipeline throughput capacity, endangering the environment, traffic safety, and even posing a serious

threat to the lives and properties of residents. Therefore, regular inspection and condition assessment of drainage pipelines, timely detection, and repair of pipeline defects can reduce or even avoid the risk of pipeline accidents and extend the life of pipelines [2,3].

Currently, in most of the urban drainage systems composed of horizontal components, pipeline inspection is mostly performed using CCTV technology, which involves entering the interior of the pipeline through a pipeline robot to record video, followed by manual defect analysis and assessment by technicians [4,5]. Although the acquisition of video data has been automated, it is still necessary for staff to identify and evaluate the defects in the video, and this method is not only less efficient in detection, but also the detection accuracy is easily affected by the experience and judgment of the inspectors. In addition, human factors may also lead to missed defects or incorrect judgments caused by visual fatigue. Meanwhile, due to the subjective judgment included in the evaluation process, different evaluators may arrive at widely varying results [6].

In the early stages of traditional computer vision, analyzing defect features in drainage pipes emerged as a key area of research. The quality of pipeline images plays a crucial role in determining the accuracy of network. Guo utilized a Zero-DCE based image enhancement method to reduce the uneven brightness, defect blurring, foggy occlusion, and smudge interference that occurs in pipeline images [7]. Halfawy employed a threshold segmentation technique to identify defective regions in pipeline images and combined the directional gradient histogram (HOG) features and SVM classifiers for defect classification, which is able to detect root defects but not for other types of defects [8]. Traditional computer vision techniques predominantly depend on manually designed features for defect recognition. This approach is not only inefficient but also struggles to meet the necessary detection accuracy requirements. The neural network technology in deep learning, which has emerged in recent years, has powerful data feature learning and processing capabilities, and thus can obtain a better representation of defect image features, showing strong generalization ability and robustness. Convolutional neural networks have been widely adopted by international researchers in the field of pipeline defect detection, leading to significant improvements in classification accuracy. Kumar developed a set of image classification systems for recognizing pipeline defects by adopting deep convolutional neural network technology. This system was able to effectively differentiate between a wide range of pipeline problems such as tree root infiltration, deposition, and rupture, and achieved a high accuracy rate of 86.2% in the experiment [9].

In advancements in AI technology, particularly the enhancement of GPU processing power, the field of real-time target detection has witnessed significant development, such as the YOLO [10–14] series and SSD [15] within one-stage neural networks. Among them, the YOLO family of algorithms is notable for its excellent performance. This single-stage detection algorithm is especially suitable for application scenarios that require high detection speed. The YOLO algorithm predicts the precise locations and categories of all targets in an image with a single forward propagation, which significantly improves the detection efficiency. While maintaining high accuracy, the YOLO series evolves by optimizing the model structure and detection strategy to adapt to changing detection environments. In addition, the YOLO algorithm demonstrates strong generalizability to effectively identify targets of different sizes and proportions while minimizing the background false detection rate. These properties have led to the widespread adoption of the YOLO algorithm in pipeline defect applications, proving its great potential in solving practical problems.

For the existing technology in the defect detection of drainage pipes, the need for human defect interpretation still exists, and current real-time and low-efficiency detection cannot accurately locate the defect location information; small defect recognition accuracy

is insufficient. At the same time, for embedded devices with limited computing power of larger computers, which is insufficient to support the computation required for large-scale inspection, model lightweighting represents a critical challenge that must be addressed during the optimization of algorithms.

To overcome these challenges, we propose the PDS-YOLO algorithm, designed for deployment in real-time pipeline defect detection systems. The key contributions of this paper are outlined as follows:

1. The paper introduces the inflated convolutional SPDSC module and the lightweight C2f-PCN module to improve the SPPF and C2f modules in YOLOv8, respectively, to reduce the feature redundancy in the spatial and channel dimensions, improve the algorithm's ability to detect rough edges of pipeline defects, and to reduce computing resource consumption. The Wise-IoU [16,17] loss function with multi-layer depth-separable convolutional MLCA [18] attention mechanism and dynamic focusing mechanism is also introduced. This approach enhances segmentation accuracy, mitigates the influence of low-quality samples on model performance, and prioritizes the extraction and representation of pipeline defect features, all without increasing the number of additional parameters or computational complexity.

2. The distributed structure of slave, host computer, and bottom controller to build a mobile embedded pipeline defect detection system is established. The optimal weights obtained from model training are imported into NVIDIA Jetson Nano, the central console of the platform, and the weight files are optimized by TensorRT to improve the inference speed of the embedded device, so that it can basically meet the demand of real-time detection of pipeline defects.

3. The structure of the paper is organized as follows: Section 2 provides an overview of the YOLOv8 algorithm, both pre- and post-optimization. Section 3 presents the pipeline defect training dataset, along with experimental results and analysis. Section 4 presents the PDS-YOLO algorithm deployed and analyzed in a mobile embedded system. In Section 5, the whole paper is summarized.

2. Materials and Methods

2.1. YOLOv8 Model

YOLOv5 has become a cornerstone in industrial intelligent detection tasks due to its exceptional modular architecture and advanced data augmentation capabilities. Building upon the YOLOv5 framework, YOLOv8 represents a next-generation one-stage object detection algorithm, offering enhanced robustness and adaptability. Tailored for industrial applications, YOLOv8 exhibits improved stability and operational simplicity, particularly in mobile and embedded environments, surpassing the usability and performance of newer iterations such as YOLOv9 and YOLOv10. YOLOv8 refers to the design of modules such as Path Aggregation Network [19], and improves on the C3 module of YOLOv5, which extracts richer information about the gradient streams for the training of the network. The YOLOv8 algorithm performs well in a variety of target detection scenarios with its powerful real-time performance and high accuracy. Similar to the method of YOLOv5 scaling coefficients, YOLOv8 is divided into five models with the same structure of n, s, m, l, and x, but with different depths, which are selected according to the arithmetic power of the device and the actual working environment. Among the YOLOv8 variants, YOLOv8n is the smallest model in terms of parameter size. While its detection accuracy is relatively lower, it offers the fastest detection speed, making it well suited for the requirements of underground pipeline defect detection systems in mobile environments. Accordingly, this paper proposes an improvement to the detection network based on YOLOv8n. The

YOLOv8 network consists of a backbone for feature extraction, a neck for feature fusion, and a detection head for calibrating the target bounding boxes, as illustrated in Figure 1.

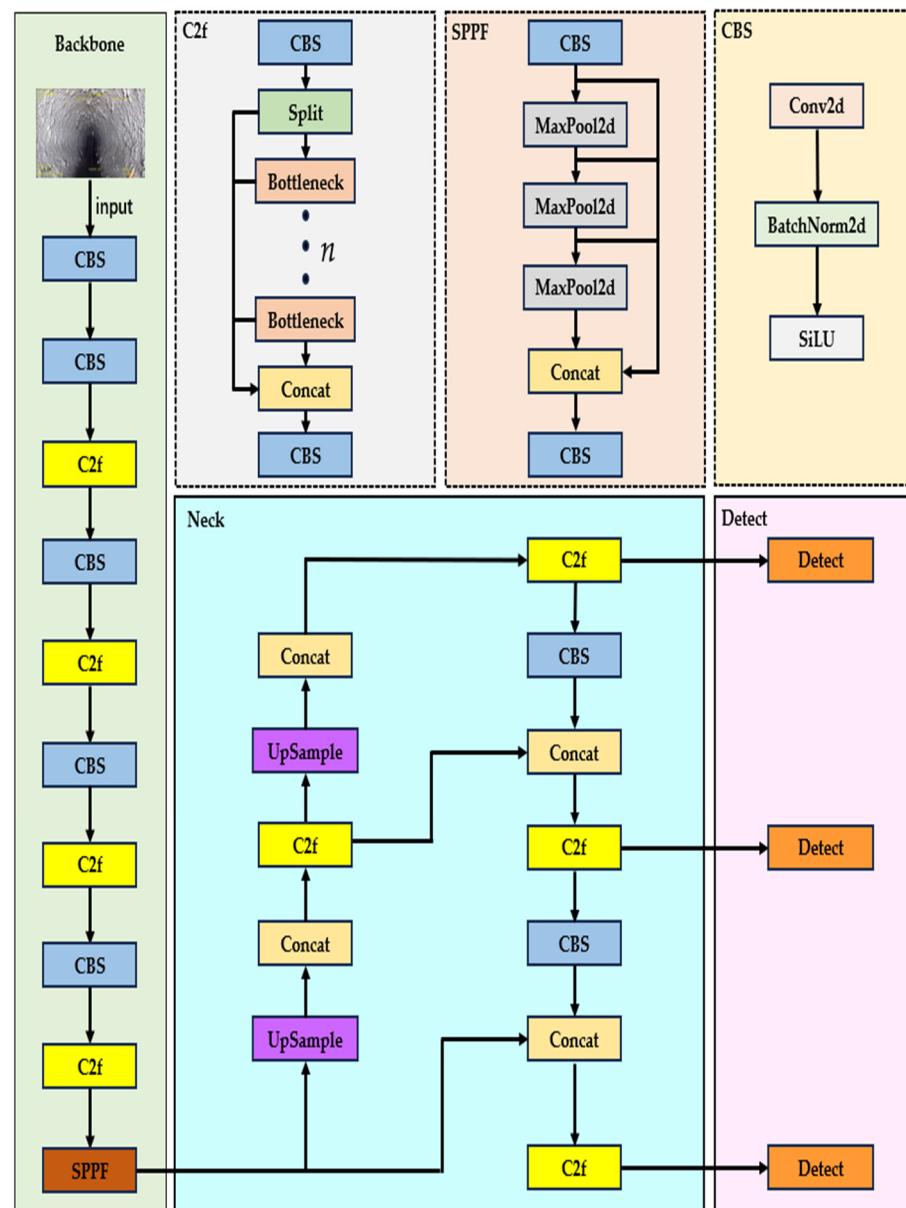


Figure 1. Network structure of YOLOv8.

2.2. PDS-YOLO Model

The PDS-YOLO pipeline defect detection and identification model constructed is illustrated in Figure 2. Initially, the C2f module in the original network is substituted with the C2f-PCN module, which retains the cross-scale feature information while reducing the computational effort through a more lightweight design. Next, the SPDSC (Spatial Pyramid Deep Shared Convolution) module replaces the SPPF module, expanding the receptive field and capturing finer-grained features, thereby improving overall model performance. To improve the network's proficiency in capturing the surface features of pipeline defects, we introduce MLCA (Mix Local Channel Attention). Finally, Wise-IoU is employed in place of CIoU as the loss function, reducing the impact of low-quality samples on model training and accelerating network convergence.

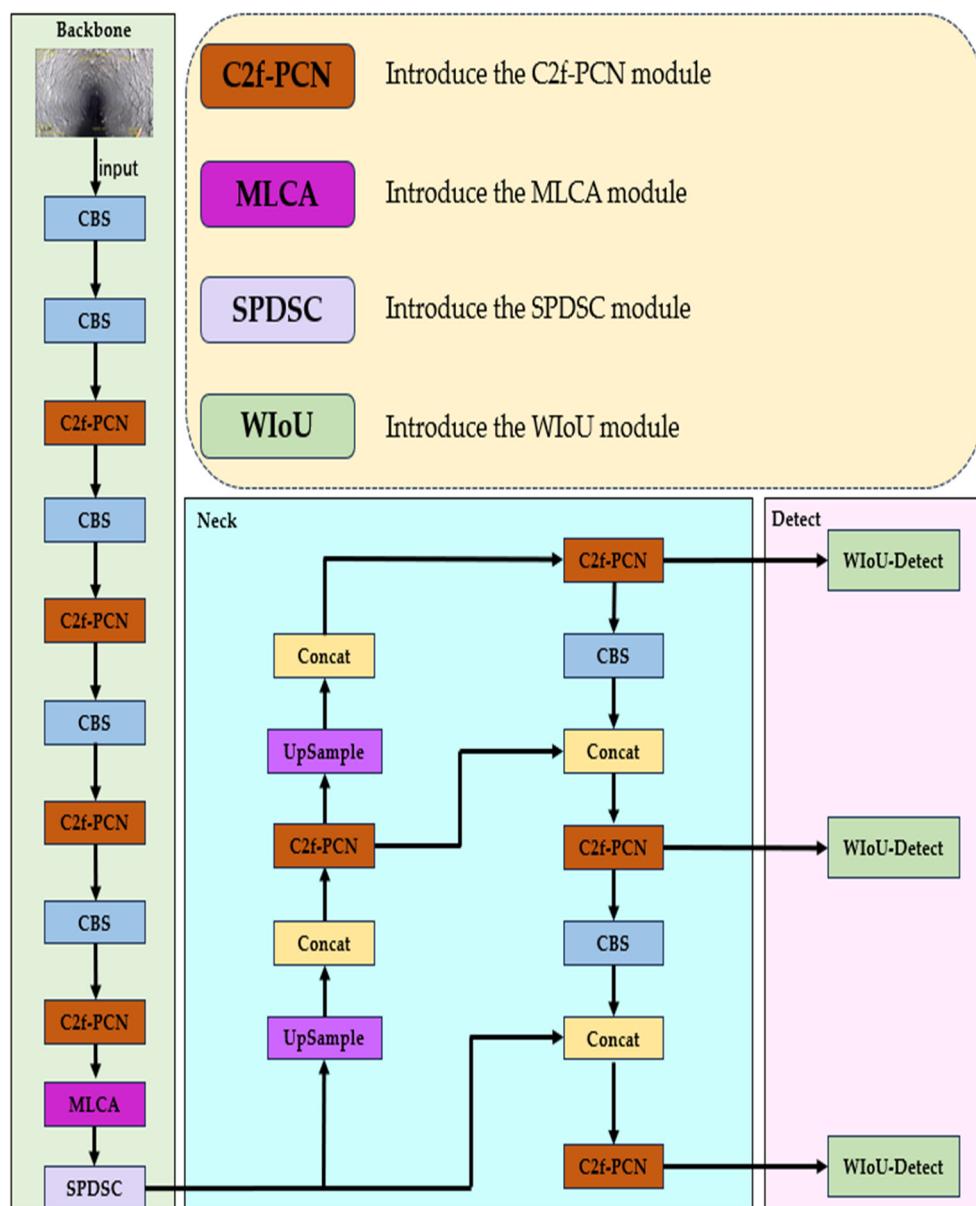


Figure 2. PDS-YOLO algorithm framework for pipeline defect detection.

2.2.1. C2f-PCN Module

In the original model, the C2f module uses a single-size convolutional kernel to extract features, and this 3×3 small-size convolutional kernel is due to the limited receptive field; the semantic features of the pipeline defect image extracted from the deeper layers of the network are relatively single, making it challenging to capture more complex, deeper features. At the same time, due to the current problem of slow inference on embedded devices due to the large amount of parameters in the pipeline defect detection task, it is still weak when dealing with complex pipeline environments and with real-time detection needs. To address these issues, we propose a new feature fusion module named C2f_PCN. In this paper, we improve the C2f module by borrowing the idea of FasterNet [20] network, and use the PCN_neck module to take the place of the Bottleneck module in the C2f. The PCN_neck makes the detection system while maintaining its lightweight and high-speed features to obtain a certain degree of performance enhancement.

FasterBlock in FasterNet is a lightweight feature extraction module in which the PConv in the module selects only a portion of consecutive channels for convolution operation to extract features, a feature that preserves a more complete and high-resolution information

transfer from shallow feature maps. Inspired by the FasterBlock feature, this paper designs the PCN_neck module as shown in Figure 3. The core function of this module is to employ convolutional kernels of three different sizes— 3×3 , 5×5 , and 7×7 —to capture information at varying scales. This approach enhances the network's feature representation capabilities and expands its receptive field, improving the overall sensory coverage. Secondly, two Chunk operations enable the network to perform convolutional operations on $1/2$ channel features and preserve the other half of the original feature information. Finally, after the Concat operation is performed on the two parts of the features, the number of channels is then adjusted by 1×1 convolution to ensure the stability of the information flow. This efficient partial convolution operation effectively preserves the original information of each level, and is also able to reduce the redundancy in both spatial and channel dimensions, significantly reducing computation and memory access.

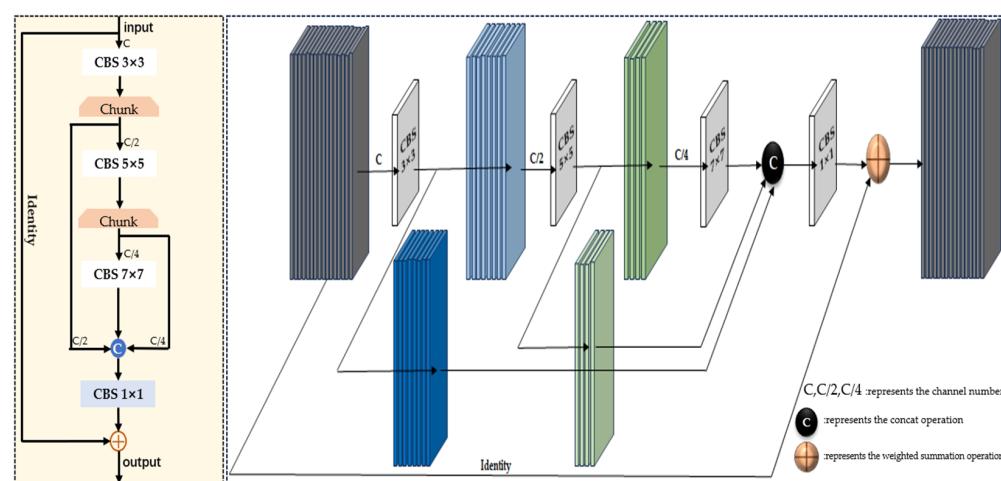


Figure 3. Structure of PCN_neck module.

2.2.2. SPDSC Module

Situated in the end of the Backbone layer, the SPPF module performs multiple maximum pooling operations on the input feature maps used to capture information at multiple scales to solve the problem of input images of different sizes. However, in the scene images where the pipeline defects are distributed far and near, and the defects vary greatly, the multi-layer pooling of the original module will cause the resolution to be reduced, and thus the detailed information of the image will be lost. To solve this problem, SPDSC (Spatial Pyramid Deep Shared Convolution) is introduced into YOLOv8 to replace the original SPPF module. The SPDSC module takes the expansion convolution as the main structure in order to enhance the sensory field size to enhance the model characterization and targeting capabilities. Each inflationary convolution operation in the module is performed in the same shared convolutional layer to achieve weight sharing of the convolutional kernel. This reduces the multi-dimensional feature redundancy and improves the computational performance of the model, thus optimizing its processing speed and resource consumption. The improved module is shown in Figure 4.

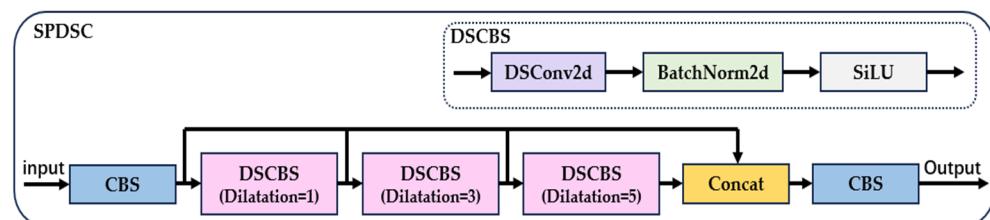


Figure 4. Network structure of SPDSC.

The module follows the concept of the hybrid inflationary convolution principle [21] by designing the DeepSharedConv-BatchNorm-SiLU (DSCBS) module, which uses the Deep Feature Shared Convolution Layer (DSConv) for feature learning, adds a batch normalization layer to accelerate convergence, and uses an SiLU activation function to increase the nonlinearity of the output features, combined with the initialized ConvModule to form the inflated convolutional processing branch. The module performs three convolution operations with different expansion rates, and outputs all the inflated convolved features by splicing them with the features that have been convolved in the first layer. The effect of extracted features with different expansion convolution rates is shown in Figure 5. This structure is designed to be less prone to losing detailed information compared to the maximum pooling operation, while being able to capture more fine-grained features. The inflated convolution has a better effect on the utilization of information from the original feature map, the pixel points involved in the computation are denser, and the gradient flow is richer.

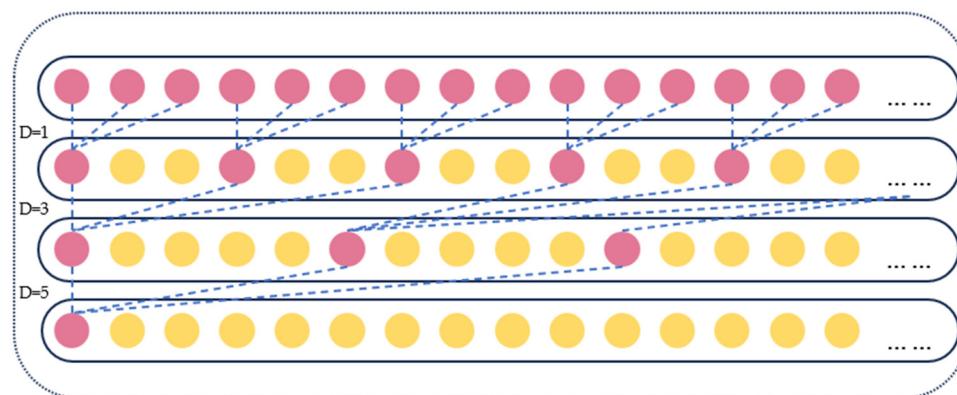


Figure 5. Extraction effect of different convolution rates.

2.2.3. MLCA Module

In general, simplifying the model's architecture can influence detection accuracy. To mitigate this issue, we introduce the MLCA [18] attention mechanism module, which is incorporated prior to the Backbone layer in the original network. MLCA is a lightweight attention module that integrates both channel and spatial information, improving the network's ability to extract information in a single environment of color information. The structure and operational principle of the MLCA module are depicted in Figure 6.

In this module, the front-end input feature map undergoes Local Average Pooling (LAP) and Global Average Pooling (GAP) to extract information from local features and the entire feature map, respectively. After that, both the locally pooled features and globally pooled features are rearranged and reorganized by one-dimensional convolution (Conv1d) on the features, and finally restored to the original feature dimensions by inverse pooling (UNAP). Among them, the rearranged local features are first processed through two fully connected layers to capture position and channel information, respectively. The position feature weights are then multiplied with the channel feature weights to compute the local hybrid channel attention weights. These weights are subsequently combined with the original input features through element-wise multiplication, effectively performing a feature selection process that extracts more discriminative features. Following this, the rearranged global features are integrated with the locally pooled features via an additional operation. This step fuses the global context within the feature map, enhancing the quality of the generated image features.

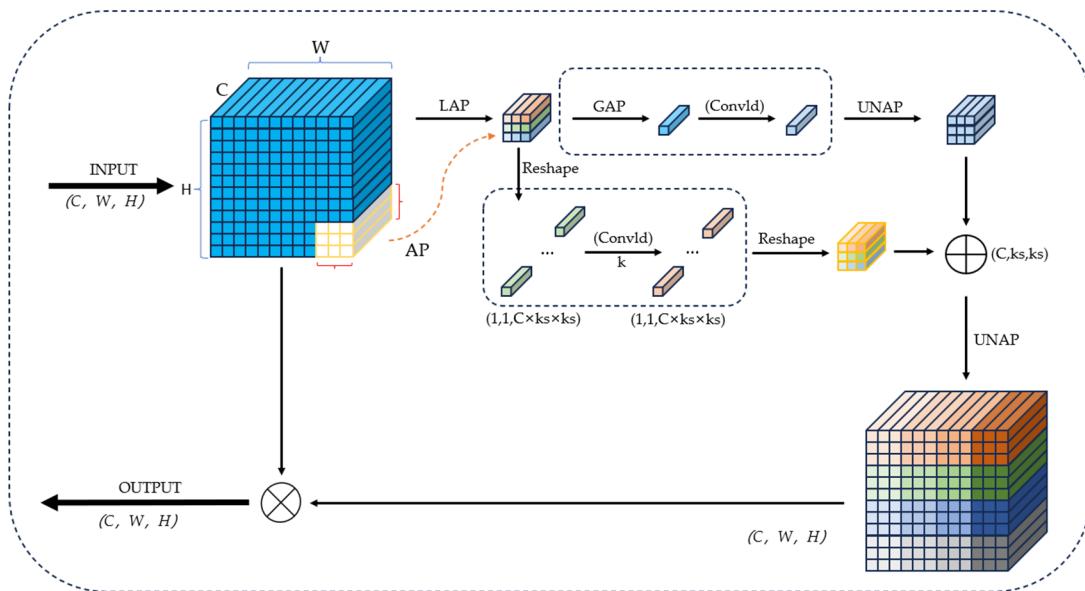


Figure 6. The structure and working principle of the MLCA network.

2.2.4. Wise-IoU Loss Function

The loss function is a fundamental building block of the pipe inspection task. In the dataset of pipeline defects, even for the same defect, the target sizes show significant diversity, and a small number of low-quality labeled boxes may appear. The loss function of the original model is CIoU, which is quantitatively evaluated by the size ratio of the target bounding box, center distance, and other factors. Because its algorithm is too cumbersome and complex, it not only spends substantial arithmetic power in the calculation process, but also the regression performance of low-quality labeled boxes will be affected.

In this paper, we use Wise-IoU (WIoU) [16,17], which is based on the dynamic focusing mechanism, as the loss function for bounding box regression, aiming to solve the negative impact that low-quality samples may have on the model performance during the training process of the pipeline defect detection model. The parameters in its closed box are shown in Figure 7. Certain early target detection algorithms, such as SSD, utilize the Hard Negative Mining strategy to balance positive and negative samples. While this approach enhances the model's ability to correctly classify difficult-to-detect samples, its relatively inefficient training process results in slower model convergence due to the inherent limitations of this strategy. Different from the traditional loss function, the core idea of WIoU is to assess the quality of anchor frames through the “outlier” metric, so as to realize a reasonable gradient assignment in a dynamic way. This improvement leads to the non-monotonic focusing learning of the network, which effectively mitigates the detrimental gradients produced by low-quality samples, thus reducing their interference in the model training, and makes the anchor frame regression more focused on ordinary quality samples. It is computed as follows:

$$\mathcal{L}_{WIoUv1} = \mathcal{R}_{WIoU} \mathcal{L}_{IoU} \quad (1)$$

$$\mathcal{R}_{WIoU} = \exp \left(\frac{\left((x - x_{gt})^2 - (y - y_{gt})^2 \right)}{\left(W_g^2 + H_g^2 \right)^*} \right) \quad (2)$$

$$\mathcal{L}_{WIoUv3} = r \mathcal{L}_{WIoUv1} \quad (3)$$

$$r = \frac{\beta}{\delta \cdot \alpha^{\beta-\delta}} \quad (4)$$

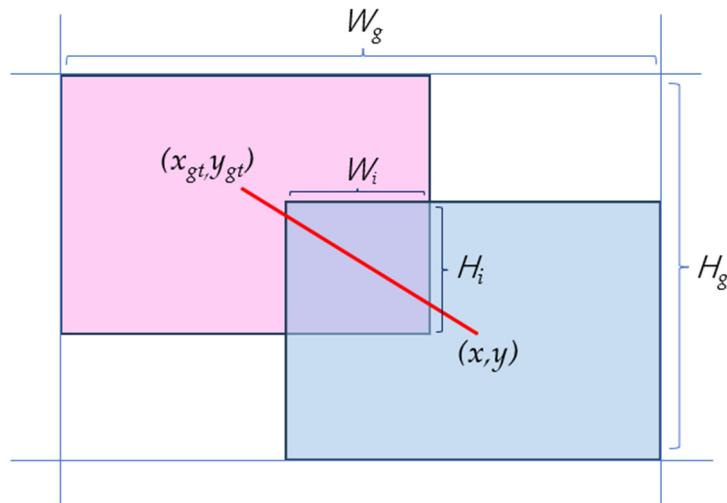


Figure 7. Wise-IoU closed-frame parameters.

In Equations (1)–(3), \mathcal{L}_{WIoUv3} represents the latest generation of Wise-IoU loss function, it is derived from the optimization of \mathcal{L}_{WIoUv1} based on a monotonic focusing mechanism. \mathcal{L}_{IoU} corresponds to the loss of IoU, and \mathcal{R}_{WIoU} denotes the distance attention. r refers to non-monotonic focusing coefficients. The coordinates x and y denote the horizontal and vertical positions of the centroid of the ground-truth frame, while x_{gt} and y_{gt} indicate the horizontal and vertical coordinates of the center of the ground-truth frame. Additionally, W_g and H_g represent the width and height of the minimum enclosing frame that bounds both the predicted and ground-truth frames. $*$ denotes the separation operation. In Equation (4), β denotes the outlier degree, which identifies the anchor frame quality; α and δ are hyperparameters, which are set to 1.7 and 2.7, respectively.

3. Experimentation and Analysis

3.1. Experimental Environment

The dataset used in this study is a combination of images collected from the CCTV detection method or QV periscopes used by SGIDI Engineering Consulting (Group) Co. in performing pipeline inspections and publicly accessible datasets from Kaggle [22]. This study focuses on a well-defined and specific sample of pipes, all of which are underground drainage systems constructed exclusively from concrete. This intentional selection ensures consistency in both material and pipeline type, enabling a more accurate and reliable assessment of the key characteristics and performance aspects of underground concrete drainage systems. In order to enrich the diversity of the image data, we employ data enhancement techniques including image flipping, cropping, brightness adjustment, saturation adjustment, and noise addition. Through this method, we can effectively expand the dataset and thus optimize the network training. The enhanced dataset comprises 3909 images. To assure the adequacy and validity of model training, the dataset is partitioned into training, validation, and test sets in a ratio of 8:1:1. This division helps mitigate the risk of overfitting, enhances the model's generalization capability, and ensures its stability in practical applications. Figure 8 shows the number of defects in each category in the dataset. The dataset was divided into six different categories: “deformation”, “scale formation”, “deposition”, “misalignment”, “obstacle”, and “rupture”. A visual example of this experimental dataset is shown in Figure 9.

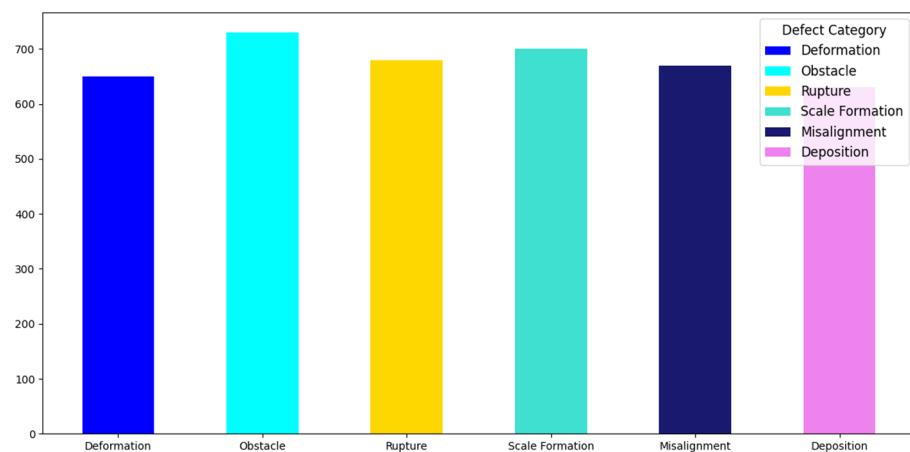


Figure 8. Number of labels for different defect types in the sample set of drainage pipe defects.

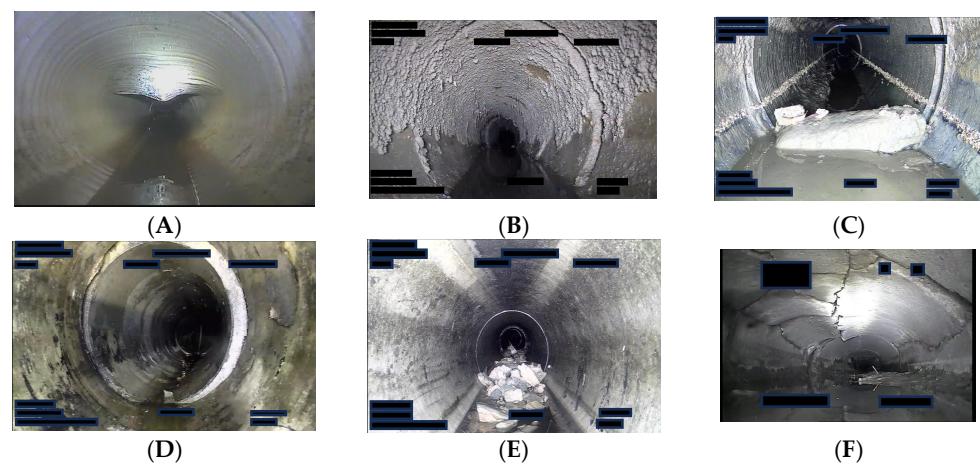


Figure 9. The six categories of the dataset: (A) deformation, (B) scale formation, (C) deposition, (D) misalignment, (E) obstacle, and (F) rupture.

In the model training experiments of this paper, the development environment is the windows environment, and the GPU used is NVIDIA GeForce RTX 4070 Super; for the rest of the platform parameter configuration, refer to Table 1.

Table 1. Computer environment and models initialization configuration.

Parameters	Configuration
CPU	AMD Ryzen 9700X
GPU	NVIDIA GeForce RTX 4070 Super
System	Windows11
Framework	Pytorch 1.8.1
IDE	VScode
Python version	3.8.19
Optimizer	SGD
Initial learning rate	0.01
Momentum	0.937
Weight decay	0.0005
Batch size	32
Epochs	200

3.2. Experimental Results and Analysis

To assess the performance of the PDS-YOLO algorithm for pipeline defect detection, we employ a set of standard evaluation metrics, including precision (P), recall (R), average precision (AP), mean average precision (mAP), F1-score, and F2-score.

Precision reflects the proportion of true positive samples among those predicted as positive by the model, while recall measures the proportion of actual positive samples correctly identified by the model. These metrics are computed as follows:

$$P = \frac{TP}{TP + FP} \quad (5)$$

$$R = \frac{TP}{TP + FN} \quad (6)$$

AP and mAP are used to compare the scores returned by the model's predicted bounding boxes and true values. They are calculated as follows:

$$AP = \int_0^1 P(R)dr \quad (7)$$

$$mAP = \frac{1}{N} \sum_{i=1}^N AP_i \quad (8)$$

The F1-score is used to comprehensively measure the model capabilities in order to equalize the effect of misdetection and omission on the model. The F2-score pays more attention to the omission, which is an important index for pipeline defect detection. Their calculation methods are as follows:

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall} \quad (9)$$

$$F2 = \frac{5 * Precision * Recall}{4 * Precision + Recall} \quad (10)$$

In Equations (5)–(8), TP, FP, and FN represent true positive, false positive, and false negative cases, respectively, while N denotes the total number of categories, and AP_i refers to the average precision (AP) of the i -th category. Additionally, this study incorporates metrics such as the model parameters, model complexity (GFLOPs), and model size. These metrics offer a comprehensive evaluation of the model's performance, capturing its efficiency and effectiveness in the detection task from multiple perspectives.

3.2.1. Ablation Experiment

In order to verify the gain effect of the existence of each improvement in the PDS-YOLO modeling strategy, multiple sets of ablation experiments are set up, and in the table “✓” denotes the point of improvement carried out for the baseline algorithm, and the specific experimental results are shown in Table 2.

Table 2. Experimental results of different improved ablation experiments on YOLOv8n.

Model	C2f-PCN	SPDSC	MLCA	Wise-IoU	P/%	R/%	GFLOPs	Parameters/M	mAP@50 (%)	mAP@50:95 (%)
Baseline					80.7	75.9	8.1	3.01	80.2	57.5
1	✓				84.1	73.5	7.1	2.60	81.2	58.3
2		✓			83.4	76.2	8.1	3.15	82.2	61.1
3			✓		85.7	76.2	8.1	3.01	82.2	57.9
4				✓	85.5	76.2	8.1	3.01	82.2	58.9
5	✓	✓			81.4	76.4	7.1	2.75	80.7	59.3
6	✓	✓	✓		83.7	79.1	7.1	2.75	82.1	59.6
7	✓	✓	✓	✓	85.5	79.9	7.1	2.75	83.6	60.1

An analysis of Table 2 shows that compared with YOLOv8, PDS-YOLO not only has better pipe defect detection performance, but also reduces the number of network parameters and thus improves the operational efficiency to a certain extent, which provides a lighter weight model for computationally constrained robotic systems. Initially, the C2f-PCN module reduces the number of parameters and GFLOPs of the model and enhances the precision and mAP of the model by a small margin. Subsequently, the SPDSC module significantly enhances the model's precision, recall, mAP@0.5, and mAP@0.5:0.95 with minimal computational overhead. Moreover, the incorporation of the MLCA attention mechanism and the Wise-IoU loss function effectively boosts defect detection accuracy without incurring additional computational costs, yielding the optimal performance metrics for the model. These results demonstrate the efficacy of the improvements introduced to PDS-YOLO in this study.

In order to further verify that the PDS-YOLO model has improved the detection performance of different categories of pipeline defects to a certain extent, the normalized confusion matrix Figure 10 and Table 3 trained before and after the model improvement are used for visualization and analysis.

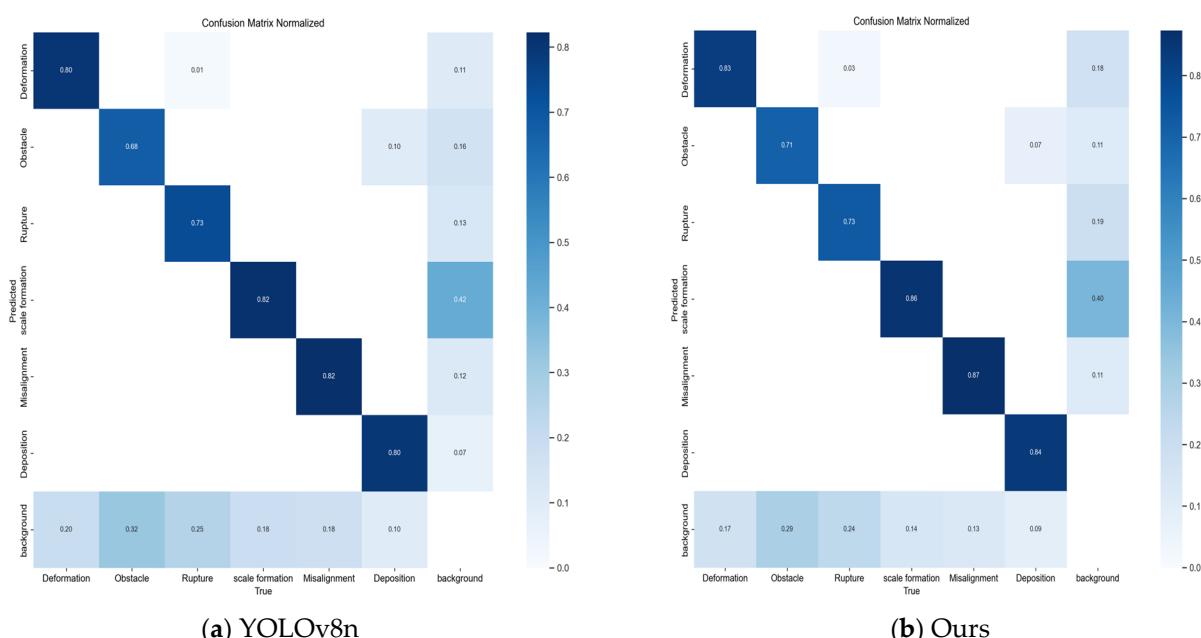


Figure 10. Confusion matrix: (a) YOLOv8n; (b) Ours.

Table 3. Experimental results of different improved ablation experiments on YOLOv8n. Performance evaluation results on the test dataset before and after model optimization.

Model	P/%	R/%	AP@50%						mAP@50 (%)
			Deformation	Obstacle	Rupture	Scaling	Misalignment	Deposition	
YOLOv8n	80.7	75.9	80.2	70.0	74.6	77.3	86.1	92.8	80.2
Ours	85.5	79.9	84.9	73.7	74.9	81.8	89.7	96.9	83.6

The diagonal values in Figure 10 represent the proportion of correct predictions made by the model, with higher values indicating better training performance. The background, treated as an implicit category within the matrix, facilitates the evaluation of the model's capacity to differentiate pipeline defects from non-defective areas. Background false positives (Background FP), presented in the matrix's final column, indicate the likelihood of the model erroneously identifying the background as a specific defect type. Conversely, background false negatives (Background FN), found in the last row, represent the proba-

bility of the model failing to detect instances in the background that should be classified as defects. Minimizing both metrics is crucial for achieving a well-trained and effective model. A comparison reveals that the improved algorithm achieves accuracy at least on par with the original algorithm across all categories. The last column and row display the miss detection and false detection rates for each category, where the improved algorithm demonstrates superior detection performance relative to the baseline. As shown in Table 3, the mAP of the enhanced PDS-YOLO model increases by 3.4%, with the AP value for the deposition category rising from 92.8% to 96.9%, reflecting an improvement of 4.1 percentage points; the AP value of the deformation category is improved from 80.2% to 84.9%, which is 4.7 percentage points; and the AP value of the scaling category is improved from 77.3% to 81.8%, an increase of 4.5 percentage points; the AP value of the obstacle category increased from 70.0% to 73.7%, an increase of 3.7 percentage points; the AP value of the misalignment category increased from 86.1% to 89.7%, an increase of 3.6 percentage points; the AP value of the rupture category increased from 74.6% to 74.9%. The AP value of the rupture category is improved from 74.6% to 74.9%, which is 0.3 percentage points, among which the improvement effect of the deformation and scaling is more obvious. Obviously, PDS-YOLO has improved accuracy for different classes and sizes of targets, and has better multi-scale detection capability.

To evaluate the effectiveness of the proposed improvements, we employ YOLOv8n and the PDS-YOLO model for visualization and comparative analysis. The final test results are presented in Figure 11. From the detected figure, it can be visualized that the scaling situation of the pipeline is complex and irregularly distributed, and YOLOv8n has a misdetection situation and missed recognition situation in the image. The improved network accurately recognizes the exact location where the scaling defects occur. At the same time, deformation may occur in the long-term operation of the pipeline. In the figure, the comparison network has missed the recognition of subtle deformation, and the improved network can completely recognize the deformation of the pipeline in the face of this situation. The enhanced algorithm proposed in this study outperforms the baseline model in detecting pipeline defects that exhibit significant variation and span different distances in complex environments. Moreover, it effectively addresses issues of misdetection and omission, which are often influenced by the subjective biases inherent in manual inspection. In the final set of comparison plots, the improved network demonstrates a higher capability to detect pipeline obstacles compared to the baseline network. However, challenges remain in accurately identifying small and distant obstacle targets. Overall, the enhanced algorithm exhibits superior localization and detection performance for pipeline defect targets. Nevertheless, further research is required to address the accurate detection of small target defects. Potential strategies include designing a specialized small-target detection layer to better utilize shallow features within the network architecture or expanding the dataset to include more examples of small pipeline defect targets.

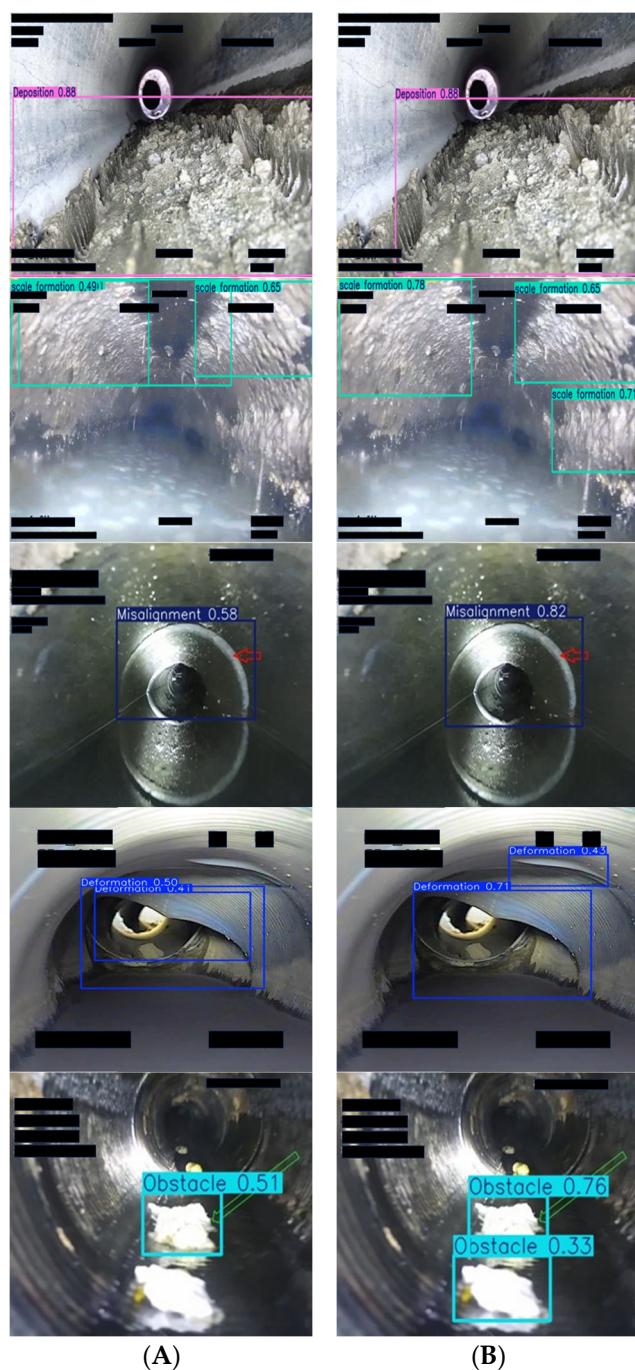


Figure 11. Comparison chart of detection effect: (A) is listed as YOLO v8 detection effect chart, (B) is listed as PDS-YOLO detection effect chart.

3.2.2. Comparison Experiments with Mainstream Benchmark Models

To substantiate the efficacy of PDS-YOLO for pipeline defect detection, we conducted comparative analyses against several prominent benchmark models, namely SSD, YOLOv3, YOLOv5, YOLOv6, YOLOv7, YOLOv9, and YOLOv10.

Table 4 shows the detection results of PDS-YOLO and SSD and other classical YOLO networks for pipe defects. In terms of detection accuracy, the PDS-YOLO model presented in this paper outperforms most models within the YOLO family and SSD across multiple metrics, including precision, recall, mAP, F1-score, F2-score, parameters, and GFLOPs. In addition, the PDS-YOLO model has only more parameters than YOLOv5n, but the accuracy is higher than this model. Experimental results demonstrate significant improvements

in both F1-score and F2-score, indicating that the PDS-YOLO model accomplishes an outstandingly good balance between accuracy and recall. Especially in the application of pipeline defect detection, which has higher requirements on recall, the F2-score, as a performance metric reflecting the importance of recall, reflects the great optimization of the model for recall, which means that the model in this paper ensures that as many positive samples as possible are detected while maintaining a high precision rate.

Table 4. Experimental outcomes compared with other models.

Model	P/%	R/%	GFLOPs/G	Parameters/M	mAP@50 (%)	mAP@50:95 (%)	F1-Score (%)	F2-Score (%)
SSD	83.9	71.6	62.7	26.2	77.1	49.8	77.2	73.8
YOLOv3-tiny	84.7	75.0	18.9	12.13	82.7	55.8	79.6	76.8
YOLOv5n	81.0	76.4	7.1	2.50	82.1	54.7	78.6	77.2
YOLOv6n	80.0	71.0	11.8	4.23	77.5	53.5	75.2	72.6
YOLOv7-tiny	82.6	72.5	13.2	6.03	79.9	50.0	77.2	74.3
YOLOv8n	80.7	75.9	8.1	3.01	80.2	57.5	78.2	76.8
YOLOv9T	82.4	73.9	7.6	2.00	80.4	56.1	77.9	75.4
YOLOv10n	81.1	70.5	6.5	2.27	76.9	52.6	75.4	72.3
Ours	85.5	79.9	7.1	2.75	83.6	60.1	82.6	80.9

From Figure 12 which displays the training results of the above comparison experiments, it is obvious that PDS-YOLO outperforms SSD and other traditional YOLO models in terms of mAP, and most of the models gradually level off after 160 cycles. After training stabilization, the mAP@0.5 of PDS-YOLO is 6.5 percentage points higher compared to the initial one-stage target detection model SSD. Compared with the new generation of YOLOv9 and YOLOv10 algorithms, the mAP@0.5 of PDS-YOLO is higher by 3.2 and 6.7 percentage points, respectively.

Various comparative experiments of PDS-YOLO

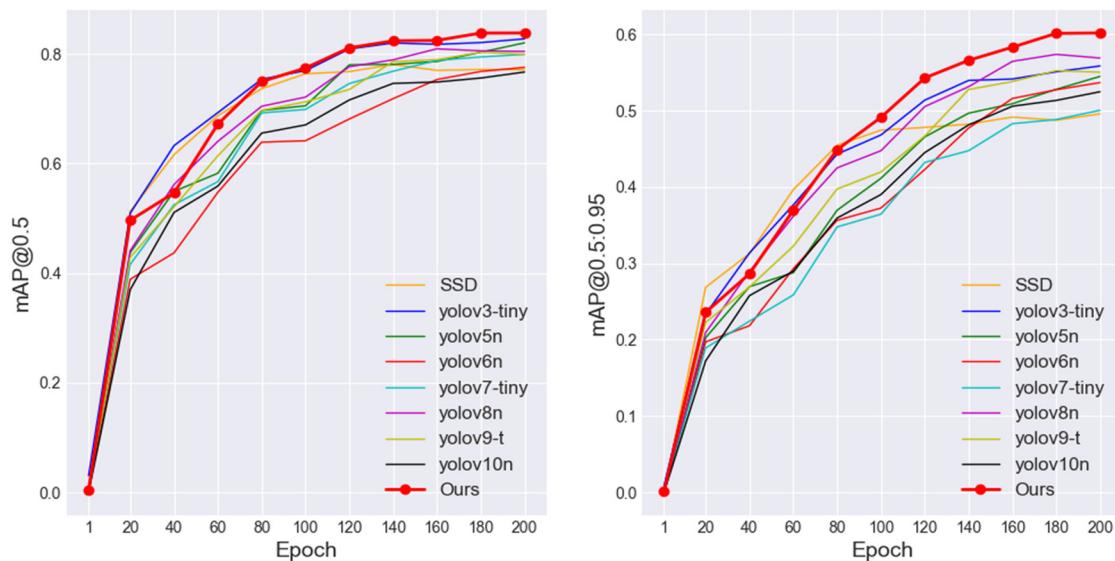


Figure 12. Comparison of mAP between different experimental models.

In summary, the PDS-YOLO model introduced in this study exhibits robust performance in pipeline defect detection tasks. Among the baseline models, YOLOv8 demonstrates superior performance in terms of mAP@0.5:0.95, surpassing all other models in this metric. While its mAP@0.5 score is slightly lower than the computationally heavier YOLOv3 and YOLOv5, it remains above average, highlighting its efficiency and effec-

tiveness. With its efficient feature extraction capabilities, PDS-YOLO achieves an optimal balance between speed and accuracy, all while maintaining a relatively low parameter count and computational cost. This exceptional overall performance underscores its effectiveness in real-world applications.

4. PDS-YOLO Model Deployment

Nowadays, there are more studies on defect target detection in drainage pipes, but most of them stay in the theoretical research stage. Especially for drainage systems consisting of horizontal structures, there are fewer algorithms that can be deployed on mobile devices. In order to realize the usability of the algorithms, in this paper, we deployed the PDS-YOLOv8n model to the central console of the mobile embedded platform prototype, the NVIDIA Jetson nano. In the original CCTV pipeline inspection method, CCTV pipeline inspection robots usually have a robust and flexible body design that can adapt to a variety of pipe diameters on horizontal structures and complex drainage pipe environments. A distributed communication framework is employed across the system's various layers, effectively simulating the functional requirements of a pipeline robot used in CCTV inspection, including tasks such as sensor data processing, environmental sensing, and wireless dual-mode communication. The integration of this system with the CCTV robot enables automated, real-time defect detection—an enhancement not present in the original approach—thereby reducing the incidence of misjudgments and omissions typically associated with manual inspection. Its basic structure is shown in Figure 13. The pipeline defect detection embedded system mainly includes the slave control terminal, the central console, the control system of the underlying components, and the image processing system. In the system, dual cameras are mounted on a multi-degree-of-freedom gimbal to take flexible pictures of the pipeline, the binocular camera is used for real-time pipeline defect detection, and the depth camera is used to collect the original, edge, point cloud, and other feature images of the defects in the pipeline.

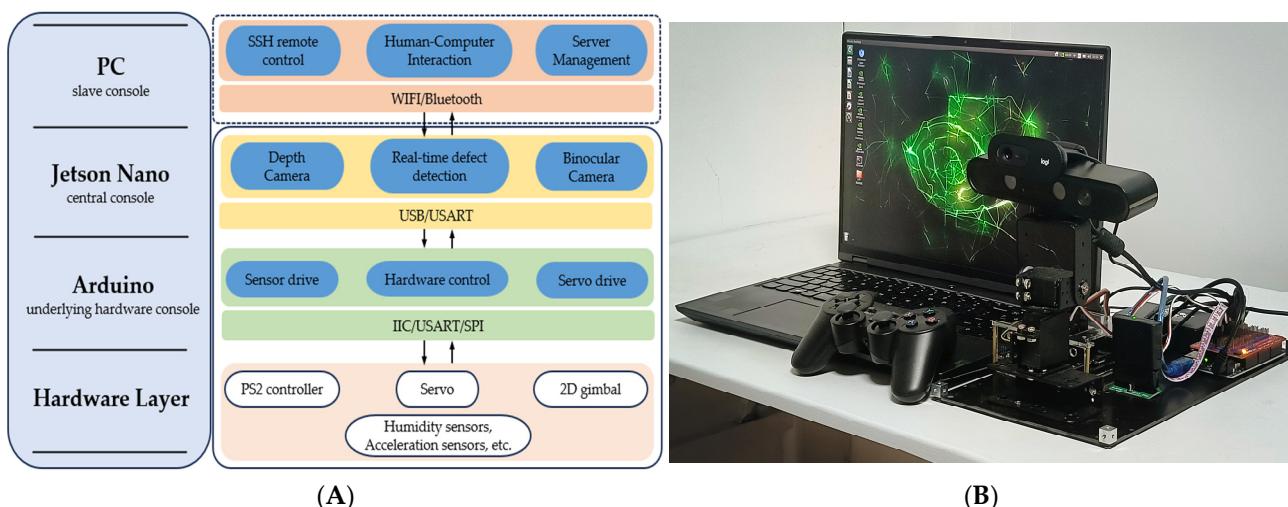


Figure 13. Mobile embedded systems: (A) is the communication framework; (B) is the testbed.

TensorRT, as the best choice for the deployment of NVIDIA series products on the ground, uses layer fusion and computational graph reshaping techniques to reduce the execution time and memory loss of the model, which greatly improves the computational efficiency and detection of real-time inference speed for embedded devices. The approach is based on the ONNX route, where first, training is performed at a deep learning host to obtain .pt weight parameter files. Then, it is converted to .onnx format in Jeston nano. onnx is an open network exchange format, and any deep learning network can be packaged

into an .onnx file, which contains the structure and parameters of the network. Finally, the .engine model file is read and optimized by TensorRT to generate a .engine model file suitable for efficient inference of the model. The .engine file is used to detect pipe defect images. The deployment process is shown in Figure 14.

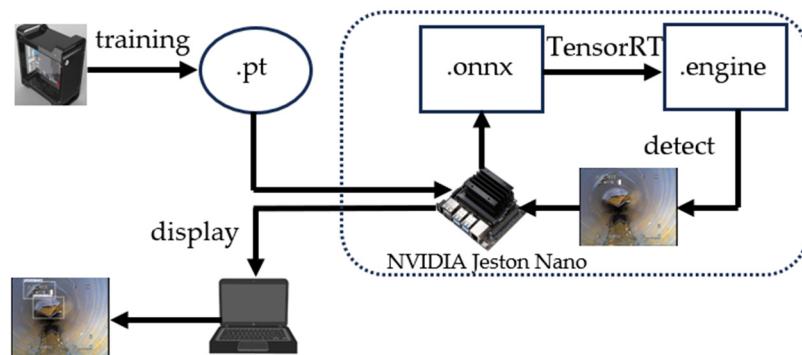


Figure 14. Model deployment process.

In order to analyze the effectiveness of TensorRT optimized deployment, this paper compares the effectiveness of the PDS-YOLOv8n model on deep learning hosts as well as on the Jester Nano, and compares the effectiveness of several parameter format files on the Jester Nano. The results are shown in Table 5.

Table 5. Comparison of model measuring effects.

Model Format	Processing Platforms	FPS (bs = 1)	mAP (%)
PDS-YOLO.pt	NVIDIA 4070 Super	190.3	83.6
YOLOv8n.pt	NVIDIA Jester Nano	4.8	80.2
PDS-YOLO.pt	NVIDIA Jester Nano	5.4	83.6
PDS-YOLO.onnx	NVIDIA Jester Nano	1.5	83.6
YOLOv8n.engine	NVIDIA Jester Nano	16.8	80.2
PDS-YOLO.engine	NVIDIA Jester Nano	19.3	83.6

As shown in Table 5, due to the strength of the GPU in the deep learning host, the model inference speed is undoubtedly the best performance. In mobile embedded devices, if the model format is deployed directly on Jester Nano without conversion, the processing speed is far inferior to that of the deep learning host, especially for real-time camera detection, the low inference speed leads to a great delay in detection that does not satisfy the demand for real-time detection, and is difficult to be applied to the actual mobile underground pipeline defect detection scenario. Utilizing TensorRT for model deployment acceleration significantly enhances inference speed, thereby demonstrating the effectiveness of the deployment process. Experimental results indicate that following the optimization of the original model file, the various performance metrics remain consistent, while the inference speed improves from 5.4 FPS to 19.3 FPS. This increase effectively satisfies the real-time detection requirements for drainage pipe defect detection in mobile scenarios. In embedded devices, the PDS-YOLO algorithm outperforms the original model in both inference speed and detection accuracy, which reflects that the algorithm is more advantageous in mobile scenarios of underground pipe inspection tasks.

5. Conclusions

In this paper, a PDS-YOLO algorithm applied to defect detection of underground drainage pipes in mobile scenarios is proposed to solve the key problems in the field of drainage pipe defect detection. The model is deployed in mobile embedded devices instead of manual real-time defect interpretation, which reduces the rate of misjudgment

and omission in the defect detection task and effectively improves the detection accuracy. In comparison to the original network, PDS-YOLO demonstrated an increase of 3.4% in mAP@0.5 and 2.6% in mAP@0.5:0.95. Additionally, this model features 8.6% fewer parameters and 12.3% lower GFLOPs than its predecessor, thereby enhancing defect detection accuracy while simultaneously diminishing computational costs and ramping up the overall efficiency of the defect detection system. In addition, the experiment also verifies that the PDS-YOLO model has a higher overall performance than other traditional inspection models. Finally, the model was deployed to the built prototype of the pipeline defect detection embedded system. The deployment results show that the optimized PDS-YOLO model can achieve high-performance real-time detection of pipeline defect images on the Jeston nano device.

To enhance the robustness of the model, we plan to expand the dataset by collecting additional pipeline defect samples across a broader range of pipeline environments, thereby extending the model's applicability. Concurrently, we are investigating alternative underground pipeline target recognition techniques and exploring model compression strategies, including pruning and distillation, to develop a more efficient and lightweight model. These improvements will facilitate the model's deployment on low-power mobile devices, such as tablets and smartphones, enabling the real-time detection of underground pipeline targets in a wider array of mobile contexts.

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

Funding: This research received no external funding.

Data Availability Statement: A portion of the defective dataset provided in this study is publicly available in the [kaggle] website, Network address: <https://www.kaggle.com/datasets/simplexitypipeline/pipeline-defect-dataset> (accessed on 12 July 2024). Another portion of the defective dataset and original contributions from this study are included in the article. For further inquiries, please contact the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest. Author Liming Zhu was employed by the company SGIDI Engineering Consulting (Group) Co., Ltd. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. Hassan, S.I.; Dang, L.M.; Mehmood, I.; Im, S.; Choi, C.; Kang, J.; Park, Y.S.; Moon, H. Underground sewer pipe condition assessment based on convolutional neural networks. *Autom. Constr.* **2019**, *106*, 12. [[CrossRef](#)]
2. Rayhana, R.; Jiao, Y.T.; Zaji, A.; Liu, Z. Automated Vision Systems for Condition Assessment of Sewer and Water Pipelines. *IEEE Trans. Autom. Sci. Eng.* **2021**, *18*, 1861–1878. [[CrossRef](#)]
3. Ali, H.; Choi, J.H. Risk Prediction of Sinkhole Occurrence for Different Subsurface Soil Profiles due to Leakage from Underground Sewer and Water Pipelines. *Sustainability* **2020**, *12*, 310. [[CrossRef](#)]
4. Hawari, A.; Alamin, M.; Alkadour, F.; Elmasry, M.; Zayed, T. Automated defect detection tool for closed circuit television (cctv) inspected sewer pipelines. *Autom. Constr.* **2018**, *89*, 99–109. [[CrossRef](#)]
5. Moradi, S.; Zayed, T.; Nasiri, F.; Golkhoo, F. Automated Anomaly Detection and Localization in Sewer Inspection Videos Using Proportional Data Modeling and Deep Learning-Based Text Recognition. *J. Infrastruct. Syst.* **2020**, *26*, 12. [[CrossRef](#)]
6. Alejo, D.; Caballero, F.; Merino, L. A Robust Localization System for Inspection Robots in Sewer Networks. *Sensors* **2019**, *19*, 4946. [[CrossRef](#)] [[PubMed](#)]

7. Guo, C.; Li, C.; Guo, J.; Loy, C.C.; Hou, J.; Kwong, S.; Cong, R. Zero-reference deep curve estimation for low-light image enhancement. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 1780–1789.
8. Halfawy, M.R.; Hengmeechai, J. Automated defect detection in sewer closed circuit television images using histograms of oriented gradients and support vector machine. *Autom. Constr.* **2014**, *38*, 1–13. [[CrossRef](#)]
9. Kumar, S.S.; Abraham, D.M.; Jahanshahi, M.R.; Iseley, T.; Starr, J. Automated defect classification in sewer closed circuit television inspections using deep convolutional neural networks. *Autom. Constr.* **2018**, *91*, 273–283. [[CrossRef](#)]
10. Farhadi, A.; Redmon, J. Yolov3: An incremental improvement. In Proceedings of the Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 1–6.
11. Wang, C.-Y.; Bochkovskiy, A.; Liao, H.-Y.M. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 7464–7475.
12. Bochkovskiy, A.; Wang, C.-Y.; Liao, H.-Y.M. Yolov4: Optimal speed and accuracy of object detection. *arXiv* **2020**, arXiv:2004.10934.
13. Nepal, U.; Eslamiat, H. Comparing YOLOv3, YOLOv4 and YOLOv5 for Autonomous Landing Spot Detection in Faulty UAVs. *Sensors* **2022**, *22*, 464. [[CrossRef](#)]
14. Qiu, X.Y.; Chen, Y.J.; Cai, W.H.; Niu, M.Q.; Li, J.Y. LD-YOLOv10: A Lightweight Target Detection Algorithm for Drone Scenarios Based on YOLOv10. *Electronics* **2024**, *13*, 3269. [[CrossRef](#)]
15. 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; Proceedings, Part I 14. Springer: Cham, Switzerland, 2016; pp. 21–37.
16. Tong, Z.; Chen, Y.; Xu, Z.; Yu, R. Wise-IoU: Bounding box regression loss with dynamic focusing mechanism. *arXiv* **2023**, arXiv:2301.10051.
17. Xiong, C.; Zayed, T.; Abdelkader, E.M.J.C.; Materials, B. A novel YOLOv8-GAM-Wise-IoU model for automated detection of bridge surface cracks. *Constr. Build. Mater.* **2024**, *414*, 135025. [[CrossRef](#)]
18. Wan, D.; Lu, R.; Shen, S.; Xu, T.; Lang, X.; Ren, Z.J. Mixed local channel attention for object detection. *Eng. Appl. Artif. Intell.* **2023**, *123*, 106442. [[CrossRef](#)]
19. 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.
20. Chen, J.; Kao, S.-H.; 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 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 12021–12031.
21. Wang, P.; Chen, P.; Yuan, Y.; Liu, D.; Huang, Z.; Hou, X.; Cottrell, G. Understanding convolution for semantic segmentation. In Proceedings of the 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), Lake Tahoe, NV, USA, 12–15 March 2018; pp. 1451–1460.
22. Pipeline Defect Dataset [EB/OL]. 2024. Available online: <https://www.kaggle.com/datasets/simplexitypipeline/pipeline-defect-dataset> (accessed on 12 July 2024).

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.