# Lightweight GCD-YOLOv5 for Real-Time Obstacle Detection in Tunnel Pipe Jacking Operations

Gang Qiao[1], Deyi Yu[1], Kun Liu[1], Qiuhan Zhao[1], Nan Yu[1], Ling Luo[2*]
[1]Power China Roadbridge Group Co., Ltd, Beijing, 100081, China
[2]School of Civil and Hydraulic Engineering, Chongqing University of Science and Technology, Chongqing, 401331, China
E-mail: luo89070989@163.com
[*]Corresponding author

**Keywords:** YOLOv5, lightweight, GCD, model pruning, knowledge distillation

**Received:** April 8, 2025

*The identification of tunnel pipe jacking obstacles is usually carried out in harsh environments, and timely and accurate identification can avoid unnecessary economic and labor losses. However, the currently commonly used obstacle recognition models are complex in structure and have long recognition feedback times. Therefore, this study proposes a tunnel pipe jacking obstacle recognition model based on an improved You Only Look Once version 5. The model is optimized through pruning and knowledge distillation techniques to enhance its lightweight characteristics and accuracy. The experiments were conducted using the COCO dataset and a custom dataset consisting of 2667 tunnel shield tunneling obstacle images. The optimized model achieved an 88.6% reduction in the number of parameters, an 84.2% reduction in floating-point operations, a 62.5% reduction in memory usage, and a 90.1% reduction in response time. In real-world testing, the model achieved an accuracy of 94.0% and a processing speed of 75 frames per second (FPS), outperforming traditional YOLOv5 and other lightweight YOLOv5 variants such as M-YOLOv5, S-YOLOv5, PL-YOLOv5, and C-YOLOv5. Using evaluation metrics such as mean Average Precision (mAP), the proposed model demonstrated high efficiency and effectiveness in real-time obstacle recognition for tunnel construction. The model provides a new technological approach for safety management in tunnel construction while improving computational efficiency and maintaining high recognition accuracy.*

*Povzetek: Lahki model GCD-YOLOv5 z Ghost in Depthwise konvolucijami, obrezovanjem in prenosom znanja omogoča bolj kvalitetno zaznavanje ovir v tunelskem potiskanju cevi v realnem času kot klasični YOLOv5 ali njegove lahke različice.*

## 1 Introduction

Tunnel pipe jacking construction technology is a commonly used non-excavation pipeline installation technique. It has advantages such as minimal environmental impact, high construction efficiency, and safety, making it widely used for underground municipal pipeline installation [1]. However, during tunnel excavation using pipe jacking technology, unforeseen obstacles often occur, which can affect construction progress, safety, and other aspects. Without accurate identification of obstacle types and appropriate handling, significant financial and material losses may occur [2, 3]. Therefore, the correct identification of obstacles during tunnel pipe jacking construction is of great importance. Currently, the main obstacle identification methods for tunnel construction include intelligent algorithm-based visual analysis, radar and multi-sensor fusion, and automated inspection robots. However, these methods require high-performance equipment and have long recognition times, which hinder timely emergency responses [4-6]. Therefore, there is an urgent need for methods that can rapidly and accurately identify obstacles in harsh construction environments. The key to obstacle identification in tunnel pipe jacking construction lies in accurately judging the size and type of obstacles. As a visual intelligence analysis algorithm, You Only Look Once version 5 (YOLOv5) can quickly and accurately analyze the multi-dimensional information of obstacle images. After being lightweight-processed, it can adapt to low-performance devices at construction sites [7]. This study innovatively optimizes YOLOv5 by replacing its overall network structure to build a tunnel pipe jacking obstacle recognition model, combining model pruning and knowledge distillation techniques for lightweight processing. This study aims to develop and validate an obstacle recognition model based on YOLOv5, which is optimized using Ghost Convolution (Ghost Conv) and Depthwise Convolutions (GCD), model pruning, and knowledge distillation, with real-world validation on tunnel construction images. Through these optimization methods, this study aims to enhance the model's computational efficiency and recognition accuracy to meet

real-time obstacle detection requirements in practical tunnel construction.

## 2 Related works

YOLOv5, with its powerful object detection and image recognition capabilities, is an upgraded version of the YOLO. It offers higher precision and faster detection speed than previous versions. Scholars worldwide have applied YOLOv5 in various fields of visual recognition. For example, Gao et al. applied YOLOv5 to detect honeycomb structures and proposed an improved YOLOv5-based honeycomb detection model using Shuffle BlockV2. This model is 92.5% faster than previous honeycomb detection models, achieves an accuracy of 96% in real-time detection, and can automatically track honeycombs within 2 seconds up to a distance of 2.5 meters [8]. Xiao's team proposed a YOLOv5-based zinc-coated steel defect detection model for metal smelting. The results of actual steel-making tests showed that the model can quickly and accurately detect surface defects of galvanized steel, and its overall performance outperforms most detection models based on other mainstream algorithms [9]. To address issues of low accuracy and high computational delays in video surveillance for ship detection, Zheng's team proposed an improved YOLOv5-based algorithm. The accuracy of the YOLOv5 model, after compression using a scaling factor, improved by 2.34%, and the detection speed reached 20 FPS, even in low-computational environments [10]. In agriculture, Chen et al. applied YOLOv5 for real-time strawberry disease detection. By incorporating the Ghost Convolution module, YOLOv5 reduced the number of parameters and floating-point operations, enhancing the model's spatial information. After adding a convolutional attention module, the model's ability to extract feature data and suppress irrelevant information was improved. The model achieved an average precision of 94.7% in test experiments [11]. For forest fire detection via visual network analysis, Zhou et al. proposed a lightweight target detection model based on YOLOv5. The model used MobileNetV3 as the backbone network within the YOLOv5 framework and was trained using semi-supervised knowledge distillation. As a result, the model size decreased by 94.1%, and the average accuracy increased by 2.6% [12].

Regarding tunnel construction obstacle recognition, several mature methods and theories have been developed and applied in practical construction. For instance, Xu et al. used drones equipped with cameras to perform automatic surface detection of tunnels to be constructed and generated target 3D shapes using a motion structure assembly line for dynamic analysis. The results showed that this method can accurately identify surface obstacles before tunnel construction [13]. Yongcan et al. addressed the issue of long identification times and high safety risks in traditional underwater tunnel obstacle recognition by proposing a robotic method that uses multi-sensor fusion for comprehensive and accurate risk analysis and evaluation of underwater tunnel construction [14]. Naranjo's team has applied intelligent algorithms to obstacle recognition and clearance in actual tunnel construction. They developed an automated obstacle recognition and clearance system for tunnel construction trucks. This system successfully identified and cleared tunnel obstacles in multiple modes, including manual and remote operation, during tests by two civil construction companies [15]. Robots with cameras are also frequently used for tunnel obstacle recognition. Li et al. reduced redundant parameters in YOLOv5 by pruning and integrated it with intelligent robots to create a high-performance tunnel fault detector. In obstacle recognition tests, the system achieved an accuracy of 81.4% and a recall rate of 98.0% [16]. Image recognition technology has also been applied to obstacle detection, typically by combining algorithms with light field cameras or other sensors to analyze feature information from images for target identification. Zhang et al. proposed an automatic classification model for tunnel rock obstacles to quickly determine rock types during actual construction. The model, based on deep convolutional neural networks trained using residual learning methods, uses extended blocks in the deep convolutional network to extract multi-scale image features. This model outperformed previous rock obstacle classification models in terms of accuracy, recall rate, and computation time [17]. The summary of the above research is shown in Table 1.

Table 1: Summary table of related works

| References | Method | Contribution | Limitations |
|---|---|---|---|
| [8] | YOLOv5 with Shuffle BlockV2 for honeycomb detection | Increased speed by 92.5%, accuracy 96%, and can track honeycombs at 2.5m in 2 seconds | Limited to honeycomb structures |
| [9] | YOLOv5 for galvanized steel defect detection | Fast and accurate defect detection, outperforming other methods | Limited to galvanized steel |
| [10] | Improved YOLOv5 with scaling factor | Improved accuracy by 2.34%, 20 FPS in low computational environments | Performance bottlenecks in complex scenes |
| [11] | YOLOv5 for real-time strawberry disease detection | Ghost convolution and attention modules, 94.7% accuracy | Limited to strawberry disease detection |
| [12] | YOLOv5 with MobileNetV3 for forest fire detection | 94.1% reduction in size, 2.6% accuracy improvement | Relies on semi-supervised learning, data quality sensitive |
| [13] | Drone-based surface | Accurate obstacle detection before | Limited adaptability to |

| | | | |
|---|---|---|---|
| | detection for tunnels | construction | dynamic environments |
| [14] | Multi-sensor fusion for underwater tunnel obstacle detection | Enhanced risk analysis and evaluation for underwater construction | Needs further optimization for complex underwater environments |
| [15] | Automatic obstacle detection and removal system for tunnel trucks | Successful multi-mode obstacle recognition and removal | Limited application to specific construction environments |
| [16] | Drone-based obstacle detection and removal | YOLOv5 pruning with high accuracy (81.4%) and recall (98.0%) | Limited to specific obstacles and scenes |
| [17] | Deep CNN-based tunnel rock classification model | Improved accuracy and recall, faster computation | Limited to tunnel rock classification |

In summary, although there has been progress in tunnel pipe jacking obstacle recognition, the methods used generally require complex technical equipment and are typically applied before construction, making them less effective for handling unexpected situations during actual construction. YOLOv5, with its fast speed, high accuracy, and low equipment performance requirements after lightweight optimization, presents an opportunity. This study proposes a tunnel pipe jacking obstacle recognition model based on lightweight YOLOv5, aiming to quickly and accurately identify obstacle types and meet the recognition needs at construction sites in practical applications.

# 3 Construction and optimization of the tunnel pipe jacking obstacle recognition model based on YOLOv5

## 3.1 Network lightweighting and model construction based on YOLOv5

YOLOv5 makes significant modifications to the original YOLO in data preprocessing and grid structures. These modifications enhance its image adaptability, detection accuracy, and generalization ability, improving its performance in visual information processing [18]. The key to obstacle recognition in tunnel pipe jacking construction is speed and accuracy. YOLOv5 can accurately predict target categories with a single forward pass. Therefore, this study leverages the advantages of YOLOv5 to construct an obstacle recognition model for tunnel pipe jacking construction. The network structure of YOLOv5 is shown in Figure 1.

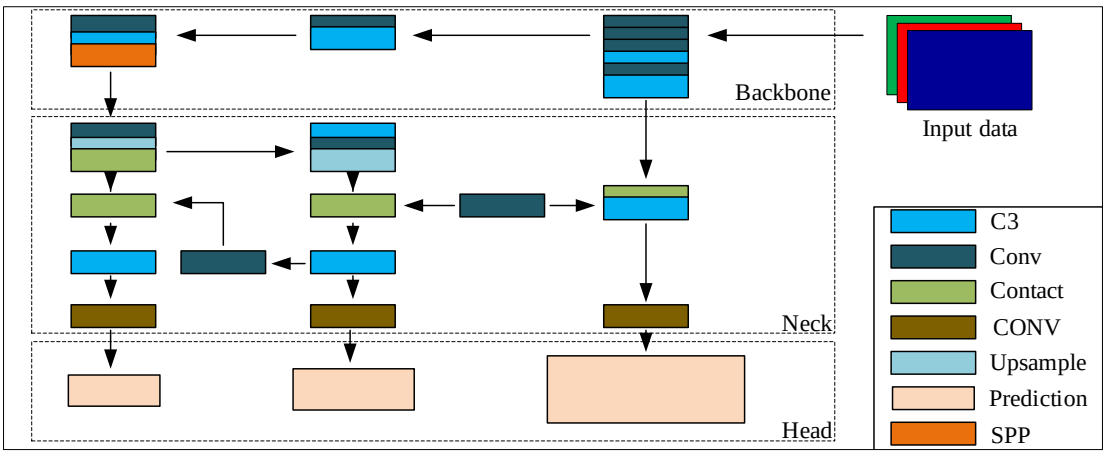



Figure 1: YOLOv5 network structure

Figure 1 presents the overall network architecture of YOLOv5. The model utilizes the Cross Stage Partial (CSP) network as the backbone and combines C3 modules and convolutional modules for feature extraction. YOLOv5 is structured into three main parts: the backbone network, the neck network, and the head network. The backbone network is responsible for extracting basic features from the input image, while the Spatial Pyramid Pooling (SPP) module further enhances feature extraction capabilities. Through residual connections and convolution operations, the backbone network fuses multiple grid cells into a more complete feature map, which is then processed in the head network for object classification and location regression. This structure ensures that YOLOv5 maintains high efficiency while performing precise object detection. YOLOv5 employs the GIOU loss function to optimize the bounding box regression, as defined in Equation (1).

$$GIoU = IoU - \frac{|A_c - U|}{|A_c|} \quad (1)$$

In Equation (1), $A_c$, $IoU$, and $U$ represent the minimum enclosing rectangle area of the detected and ground truth boxes, the Intersection Over Union (IoU), and the union area, respectively. YOLOv5s adopts CSP as

its backbone network. However, CSP networks contain numerous C3 and CONV modules, leading to excessive redundant channels and data, which increases the overall computational and parameter complexity of YOLOv5 [19]. In real-world tunnel pipe jacking construction, a lightweight and fast network structure is necessary for obstacle recognition. To achieve this, this study replaces the CSP network with a lightweight alternative while ensuring network stability. The proposed solution employs a Depthwise Convolution (DWC) network to achieve inverse residual concatenation in a Ghost Convolution (GC) network, forming the Ghost Convolution-Depthwise Convolution (GCD) network as a replacement. The construction process of the GCD network is shown in Figure 2.

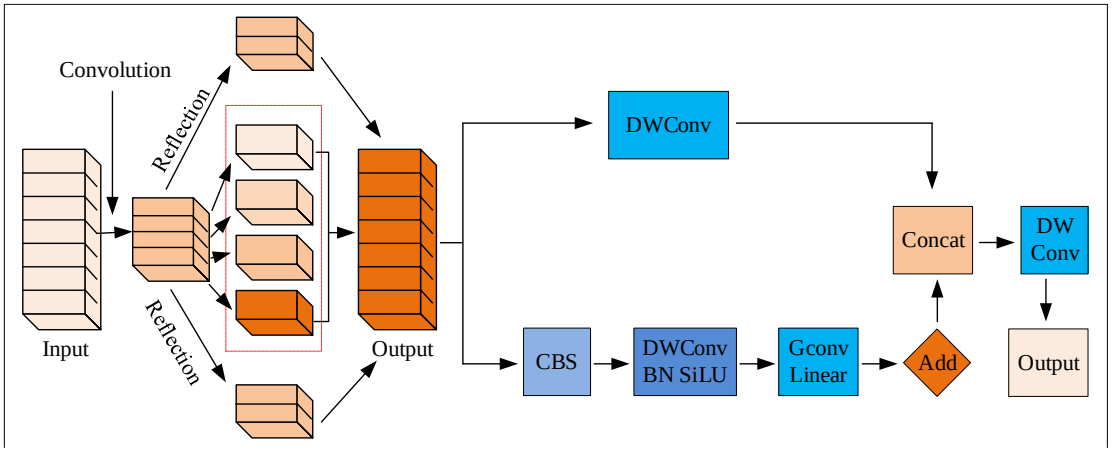



Figure 2: Flowchart of constructing a phantom convolutional depthwise separable network

In Figure 2, the core part of the GC network is implemented through the DWC layer and the inverse residual concatenation. Specifically, the red box in the figure indicates the implementation method of residual connections in the network. In this part, the input feature map first undergoes several layers of convolution operations (such as CBS and DWC), which extract image features and perform feature processing. Next, through the addition operation, the input feature map is added to the output feature map of the convolutional layer. This is the core process of inverse residual joining. Through this operation, the network can retain the information of the input features while avoiding information loss during the feature extraction process. Finally, the output feature map is concatenated with other features through the Concat operation, and further processed through the DWC and Linear layers to obtain the final output. In the GC structure, the Ghost Convolution process applies multiple convolution operations on the original image to extract real feature maps while generating ghost feature maps. The calculation process is defined in Equation (2).

$$y_{ij} = \phi_{ij}(y'_i), \Lambda_j = 1,...,s \quad (2)$$

In Equation (2), $y_{ij}$ represents the generated ghost feature map, $y'_i$ denotes the feature in the ghost layer, and $\phi_{ij}$ represents the $i$ -th layer's $j$ -th linear operation. $s$ is the maximum number of feature layers. When performing linear combinations on real feature maps, the convolution ratio between layers is shown in Equation (3).

$$r_s = \frac{n*c*k*k*h'*w'}{n/s*h'*w'*k*k+(s-1)*n/s*h'*w'*d*d} = \frac{s*c}{s+c-1} \approx s \quad (3)$$

In Equation (3), $n$ represents the number of parameters, $h'$, $w'$, and $k$ represent the height, width, and depth of the final feature map, respectively. Finally, the ghost and real feature maps are linearly combined, with the parameter convolution ratio during computation shown in Equation (4).

$$r_c = \frac{n*c*k*k}{n/s*c*k*k+(s-1)*n/s*d*d} \approx \frac{s*c}{s+c-1} \approx s \quad (4)$$

In Equation (4), $r_c$ represents the final scaling factor. Part of the feature maps undergoes depthwise separable convolution after GConv processing, while the rest are processed through Conflict-Based Search (CBS) in a Batch Normalization (BN) and SiLU-optimized DWConv network for inverse residual learning. These two sets of feature maps are then fused in the connection layer. Replacing the CSP network in YOLOv5 with the GCD network results in the GCD-YOLOv5 model for tunnel pipe jacking obstacle recognition, as shown in Figure 3.

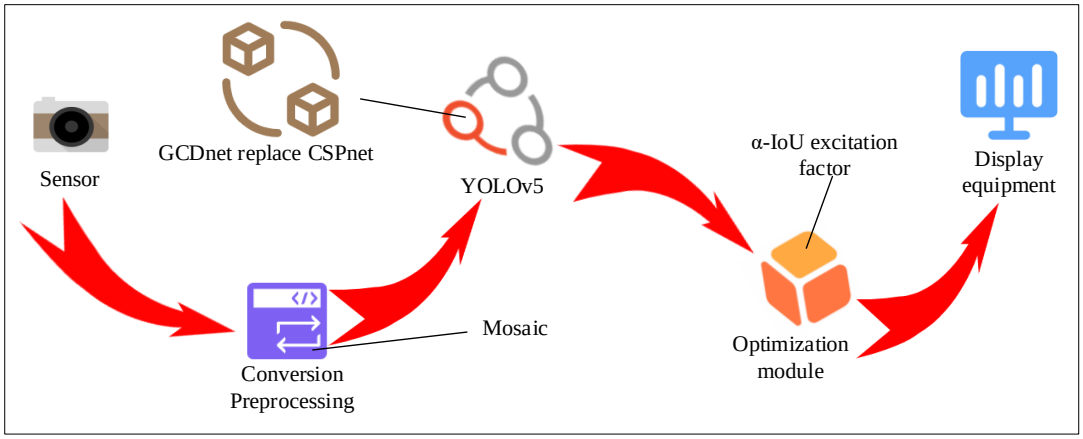



Figure 3: GCD-YOLOv5 obstacle recognition model operation flow chart

As shown in Figure 3, the model first captures obstacle images using cameras or other sensors. Before entering the GCD-YOLOv5 network, Mosaic preprocessing is applied, including image scaling, cropping, and concatenation. The preprocessed images are then fed into the GCD-YOLOv5 network for convolution processing. The processed data is refined using the a-IOU function, as defined in Equation (5).

$$L_{a-DIoU}(A,B) = 1 - IOu(A,B)^{a} + \frac{\rho^{2a}(A,B)}{c^{2a}} \quad (5)$$

In Equation (5), $L_{a-DIoU}$ represents the loss function incorporating a-IOU into the original GIOU design. $A$ and $B$ denote the target box and predicted box sizes, respectively. $\rho$ is the Euclidean distance between the centers of the two boxes, while $c$ represents the diagonal length of the minimum enclosing rectangle containing both boxes. The optimized feature maps serve as the model's final prediction results and are displayed on output devices. For tunnel pipe jacking obstacle recognition, the model promptly identifies obstacle types and provides relevant information.

## 2.2 Optimization of GCD-YOLOv5 using model pruning and knowledge distillation

The GCD-YOLOv5 model achieves lightweight optimization through network structure replacement, improving accuracy. However, further lightweight optimization is needed for real-world tunnel pipe jacking construction. Network pruning assigns different weights at the channel, kernel, and layer levels, reducing the model's memory footprint without additional training [20]. The network pruning process for the GCD-YOLOv5 model is shown in Figure 4.

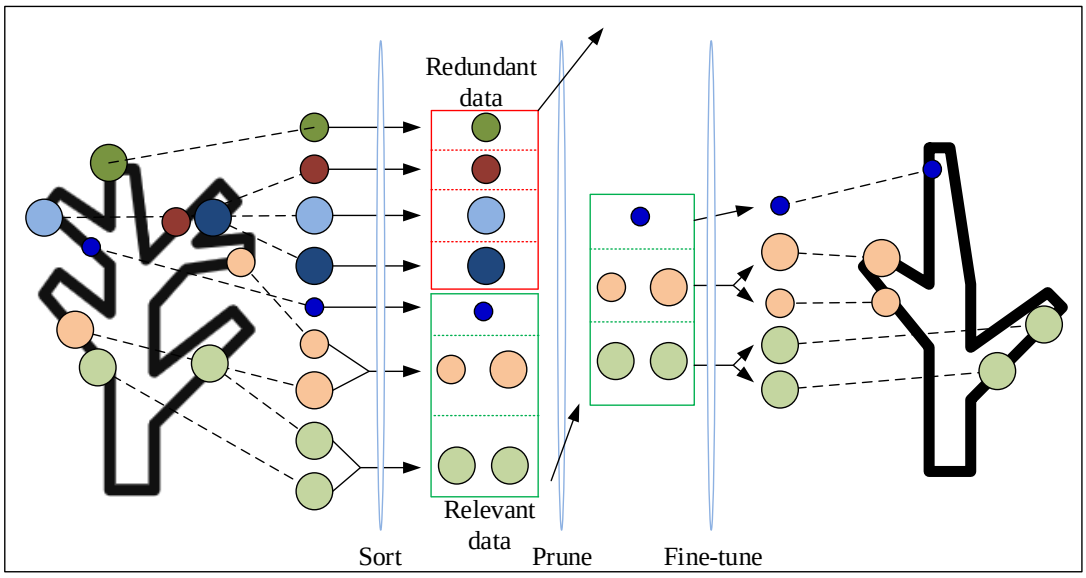



Figure 4: Schematic diagram of network pruning process

As shown in Figure 4, network pruning integrates data from each channel in the GCD network, classifying it as either feature or non-feature data. Only feature data is retained and reassigned to new network channels. However, uncontrolled channel data during pruning may result in mismatched sizes and distributions between predicted and target images, significantly reducing accuracy [21]. To address this issue, BN is applied to align

channels, as defined in Equation (6).

$$x'_i = \frac{x_i - E[x]}{\sqrt{Var[x]}} \quad (6)$$

In Equation (6), $x'_i$ represents the final data, $x_i$ is the input data, $E[x]$ is the mean, and $Var[x]$ is the variance. To prevent data homogenization, the scaling factor $\gamma$ and offset $\beta$ are applied to adjust the output data, as shown in Equation (7).

$$y_i = \gamma x_i + \beta \quad (7)$$

In Equation (7), $y_i$ represents the scaled output data. Each convolution channel, depending on $\gamma$, switches between open and constrained states, with constrained channels being pruned. During the specific pruning process of the model, the study set that 30% of the neurons or channels would be pruned each time during the pruning operation to ensure an effective reduction in the network size while maintaining good model performance. Pruning decisions are based on the weight magnitude of each neuron or channel. A minimum weight threshold of 0.005 was set. Neurons or channels below this threshold will be pruned to ensure that the pruning process has a relatively small impact on the model output. The pruning strategy adopts channel-based pruning, that is, pruning by channel, which can reduces redundant calculations more precisely. To ensure the validity of the model, a 5% performance tolerance is set. That is, the accuracy of the model after pruning shall not decrease by more than 5%. If the accuracy decrease caused by pruning exceeds this threshold, the pruning operation will be revoked. The pruned model has fewer channels and parameters, leading to lower accuracy. Knowledge distillation transfers information from a trained large model to a smaller model to improve accuracy [22]. Thus, the unpruned GCD-YOLOv5 model serves as the teacher model for knowledge distillation, as shown in Figure 5.

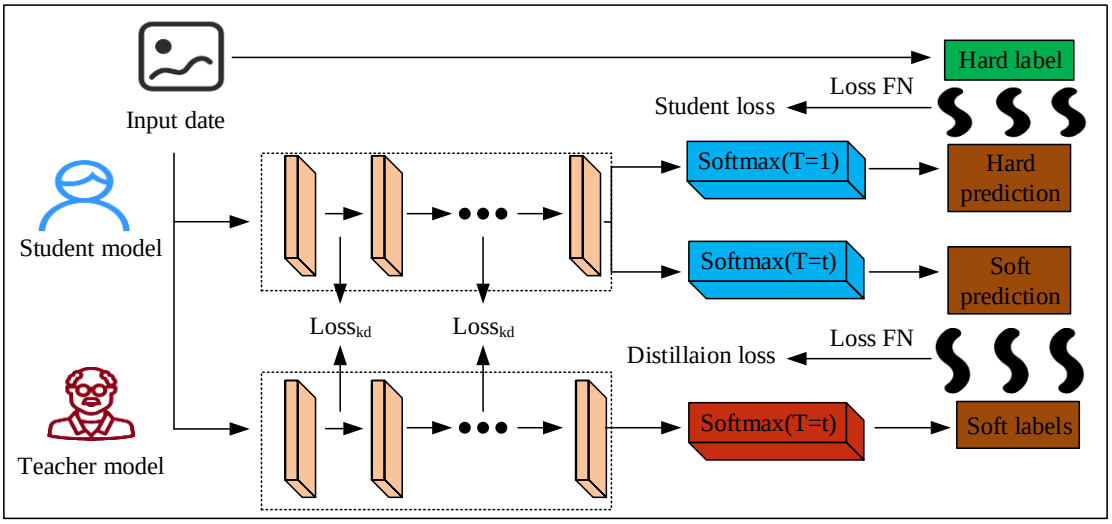



Figure 5: Schematic diagram of the knowledge distillation process

As shown in Figure 5, knowledge distillation requires both the teacher and student models to process the same input data and extract similar feature sets. Weighted cross-entropy loss addresses the difference in feature weighting between the two models. To enhance classification ability, the study employs a Region Proposal Network (RPN) and a Recursive Cortical Network (RCN) for learning. To maintain the lightweight feature of YOLOv5, the introduction of RPN and RCN is limited to the training stage, serving as knowledge transfer tools between the teacher model and the student model. Specifically, RPN is used to generate "soft" candidate regions (i.e., candidate box proposals), which are provided by the teacher model and passed to the student model as auxiliary information during the training process to help it learn the spatial distribution of the target region and thereby optimize the classification ability. Similarly, RCN enhances the classification ability in the student model, extracts more distinguishable features through recursive learning, and improves the recognition ability for multi-category targets. It should be noted that RPN and RCN do not directly participate in the inference process of YOLOv5. They only act on the student model during the training stage, helping the student model improve detection accuracy and classification ability through knowledge distillation without affecting the inference speed. Therefore, although these two components are different from the traditional YOLOv5 architecture, they ensure through auxiliary optimization that the model can achieve high object detection accuracy and classification ability while maintaining lightweight. The definitions of the two are shown in Equation (8).

$$\begin{cases} LRCN = \frac{1}{N}\sum_i L_{cls}^{RCN} + \lambda \sum_j L_{reg}^{RPN} \\ LRPN = \frac{1}{M}\sum_i L_{cls}^{RPN} + \lambda \frac{1}{M}\sum_j L_{reg}^{RPN} \end{cases} \quad (8)$$

In Equation (8), $N$ and $M$ represent RCN and RPN processing numbers, respectively. $\lambda$ is the scaling factor, $L_{cls}$ represents the loss between the student model's predicted and actual values, and $L_{reg}$ represents

the loss between the student and teacher models. The final student model output loss is given in Equation (9).

$$L = L_{RCN} + L_{RPN} + \lambda L_{H\,\mathrm{int}} \quad (9)$$

In Equation (9), $L_{H\,\mathrm{int}}$ is the correction function based on hidden layers. $L_{cls}$, $L_{reg}$, and $L_{H\,\mathrm{int}}$ continuously adjust the student model to match the teacher model's predictions. During training, when the student model outperforms the teacher model, the results are corrected, as shown in Equation (10).

$$L_{reg} = L_{sL1}(R_s y_{reg}) + vL_b(R_s R_t y_{reg}) \quad (10)$$

In Equation (10), $y_{reg}$ represents ground truth labels, $v$ is the weight coefficient, $R_s$ and $R_t$ are the student and teacher model outputs, $L_b$ represents teacher regression constraints, and $L_1$ is the smooth loss between student predictions and actual results. When evaluating final outputs, hidden layer distances are used for representation, as defined in Equation (11).

$$L_{H\,\mathrm{int}}(VZ) = \left\| V - Z \right\|_2^2 \quad (11)$$

In Equation (11), $V$ and $Z$ represent the outputs of the guided student model and intermediate layers of the teacher model, respectively. The final student model output can also be expressed using smooth loss on $L_1$, as shown in Equation (12).

$$L_{H\,\mathrm{int}}(VZ) = \left\| V - Z \right\|_1 \quad (12)$$

In Equation (12), $L_{H\,\mathrm{int}}$ measures the quality of the final output. The trained student model, despite having fewer parameters than the teacher model, achieves high recognition accuracy. Ultimately, the lightweight GCD-YOLOv5 model, optimized through pruning and knowledge distillation, is shown in Figure 6.

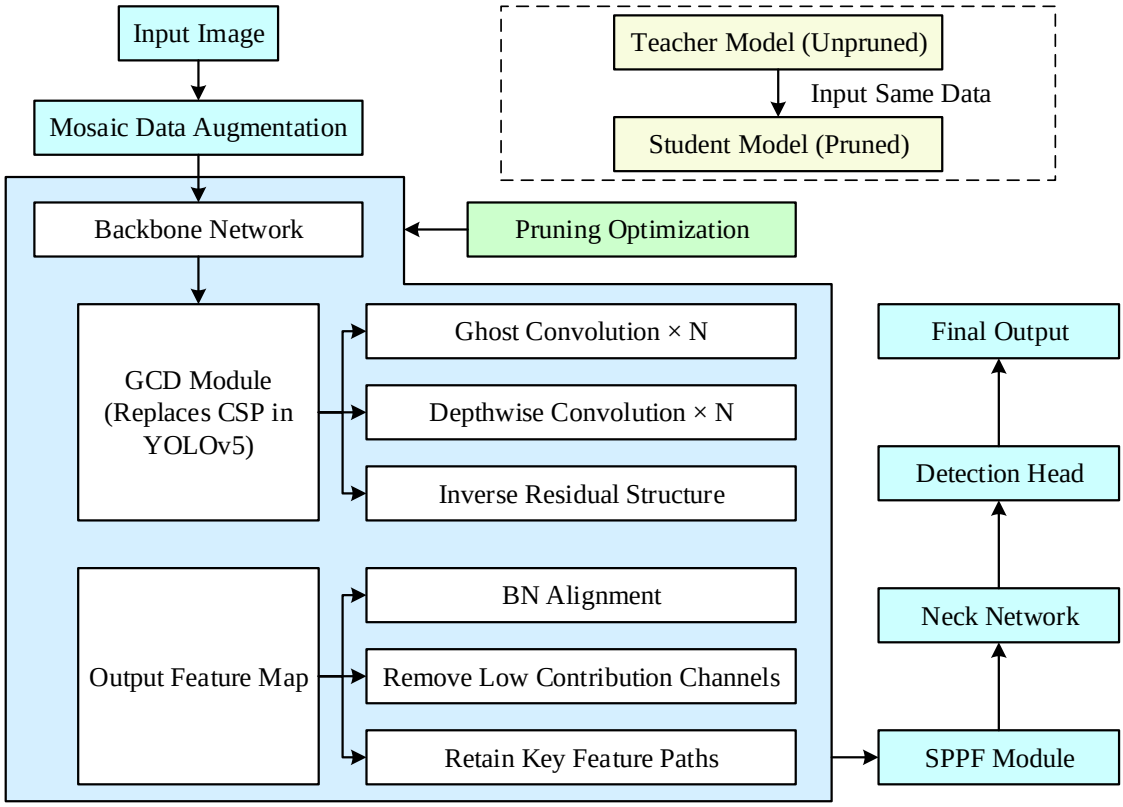



Figure 6: The lightweight GCD-YOLOv5 model

In Figure 6, the structure of the final lightweight GCD-YOLOv5 model improves computational efficiency and recognition accuracy through a series of optimization measures. This model first adopts Ghost convolution and deep convolution to replace the CSP network in YOLOv5, reducing computational complexity while maintaining the efficient feature extraction capability. Then, pruning optimization ensures the removal of redundant channels and convolutional layers without affecting performance, and stability after pruning is guaranteed through BN alignment. The optimized model transfers the knowledge of the teacher model to the student model through knowledge distillation, which can further improve the accuracy rate. Therefore, the GCD-YOLOv5 model after these optimizations can effectively adapt to the requirements of rapid object detection in actual scenarios such as tunnel pipe jacking construction.

# 4 Validation of the pipe jacking obstacle recognition model based on lightweight YOLOv5

## 4.1 Performance validation of the obstacle recognition model

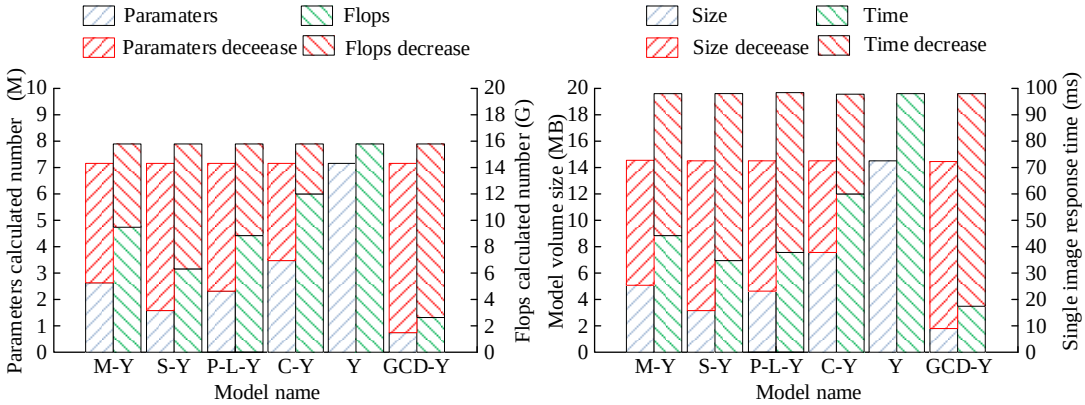
To evaluate the performance of the lightweight GCD-YOLOv5 model, the study tested the model using the Common Objects in Context (COCO) dataset. This dataset

is a standard dataset widely used for object detection, instance segmentation, and keypoint detection. It contains 80 object categories and hundreds of thousands of images, providing detailed annotation information such as bounding boxes and semantic segmentation masks. During training, the number of epochs was set to 50, the batch size to 32, the learning rate to 0.001, and the Adam optimizer was used. To ensure the reliability and robustness of the results, all models were independently trained five times, and the average of the results was taken to reduce the influence of accidental factors. In addition, data augmentation techniques such as random clipping, rotation and horizontal flipping were adopted during the training process to improve the generalization ability and adaptability of the model. The M-YOLOv5 model based on MobileNetV2, the S-YOLOv5 model based on ShuffleNetV2, the P-L-YOLOv5 model based on PP-LCNet, the C-YOLOv5 model based on C3Ghost, and the original YOLOv5 model were used as comparisons. Among them, M-YOLOv5 adopts MobileNetV2 as the backbone network to reduce the computational calculation [23]. S-YOLOv5 is based on ShuffleNetV2, further reducing the computing resource consumption and was suitable for low-computation environments [24]. PL-YOLOv5 adopts PP-LCNet to optimize the backbone network and improve accuracy and efficiency [25]. C-YOLOv5 combines the C3Ghost backbone network and reduces memory usage and computational complexity through techniques such as pruning [26]. The original YOLOv5 serves as the benchmark model, providing comparisons with other optimized variants. Through the comparison of these variants, the impact of different optimization strategies on the model performance can be comprehensively evaluated, helping to select the most suitable scheme for the obstacle identification task in tunnel construction. The test configuration is shown in Table 2.

Table 2: Specific configuration for the experiment

| Hardware environment | | Software environment | |
|---|---|---|---|
| Test environment | Specifications | Test environment | Specific parameter |
| CPU | Intel(R) Xeon(R) Silver4215 | CUDA | 10.2 |
| GPU | NVDIA GTX 1080Ti | CUDNN | 8.0.4 |
| Operating system | Windows11 | Programming language | Python |
| Memory size | 64G | Edition | Anaconda 3-5.2.0 |

All models were tested under the same hardware and software environment. To verify whether the proposed model had advantages in lightweight characteristics, the floating-point operations and model size of each model were compared. The results are shown in Figure 7.




(a) Each model parameter and floating-point size diagram (b) Each model size and response time diagram

Figure 7: Comparison of lightweight indicators of various models

As shown in Figure 7(a), the proposed GCD-YOLOv5 model has the lowest number of parameters and floating-point operations, with only 0.8M±0.02M parameters and 2.5G±0.1G FLOPs, significantly outperforming the original YOLOv5 model and other optimized variants. Compared to the unoptimized model, the number of parameters was reduced by an average of 88.6% (95% CI: [87.9%, 89.3%]), and the FLOPs were reduced by 84.2% (95% CI: [83.5%, 84.8%]), confirming the model's efficiency in memory utilization during runtime. As shown in Figure 7(b), among the six models, GCD-YOLOv5 achieved the smallest memory footprint at 1.8MB±0.05MB and the shortest response time at 19ms±0.8ms per image. Compared to the original YOLOv5, memory usage was reduced by 62.5% and response time by 90.1%. All models were tested across

five independent runs, and paired t-tests were applied to response time and memory usage. Results confirmed that the proposed model demonstrates statistically significant improvements in both metrics ($p < 0.01$), validating its lightweight performance advantages. To further evaluate the model's performance, the study analyzed the relationship between training epochs and accuracy, as shown in Figure 8.

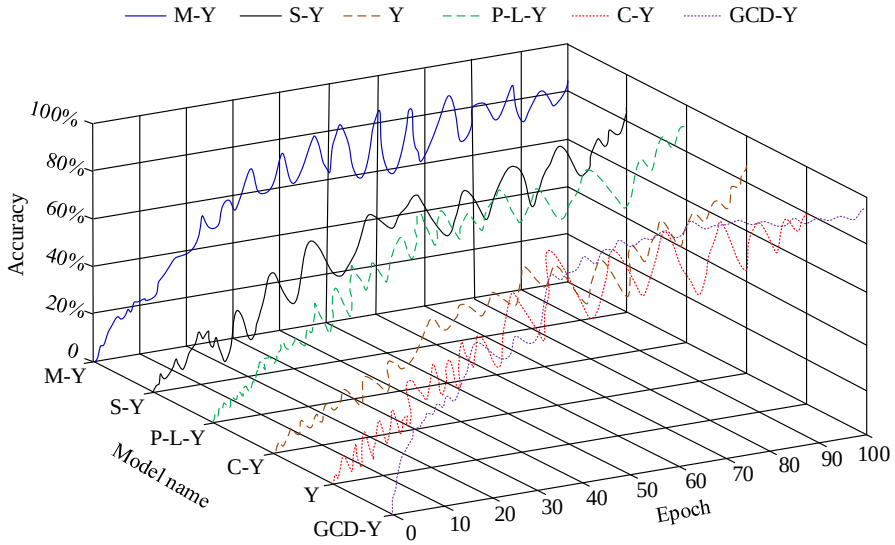
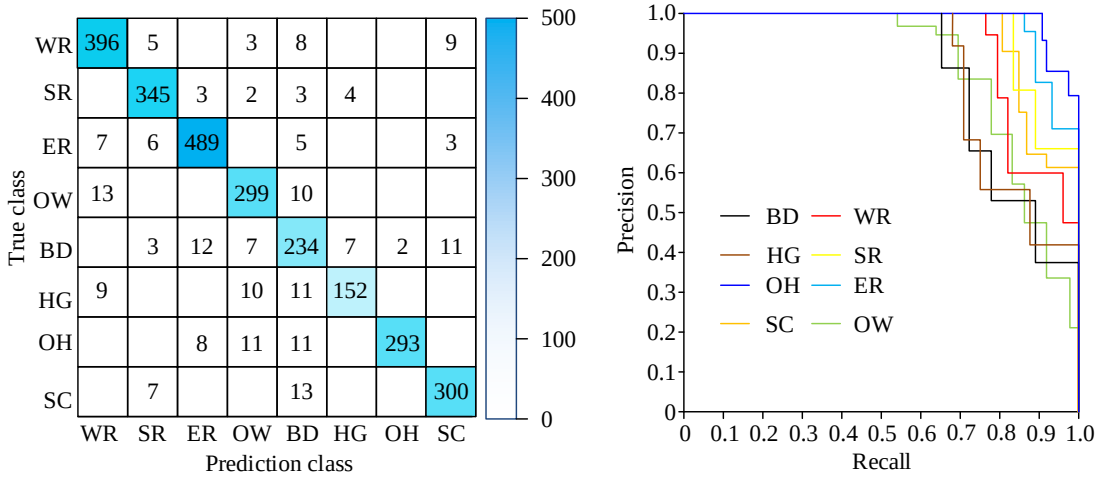



Figure 8: Comparison of accuracy changes

As shown in Figure 8, the accuracy of all models increases with the number of training epochs. At epoch 10, the accuracy of M-YOLOv5, S-YOLOv5, PL-YOLOv5, C-YOLOv5, and the original YOLOv5 was 20.2%, 18.4%, 18.6%, 13.1%, and 18.2%, respectively. In contrast, the proposed GCD-YOLOv5 achieved 32.5%±1.2% accuracy at the same stage, significantly outperforming the others, indicating faster learning capability. After 100 epochs, the final accuracies were 87.0%±0.6% (M-YOLOv5), 86.2%±0.7% (S-YOLOv5), 90.1%±0.5% (PL-YOLOv5), 89.5%±0.6% (C-YOLOv5), and 80.2%±0.8% (YOLOv5), while the proposed GCD-YOLOv5 achieved the highest at 93.5%±0.4%. The 95% confidence interval for the GCD-YOLOv5 final accuracy was [92.9%, 94.1%], and paired t-tests showed statistically significant differences compared with all other models ($p < 0.01$). These results demonstrate that the proposed model offers both faster convergence and higher accuracy.

## 3.2 Practical application of the obstacle recognition model

After verifying the lightweight characteristics and recognition performance of the GCD-YOLOv5 model, the study further tested its performance in real-world applications using 2,667 images of tunnel pipe-jacking construction obstacles collected from actual construction processes. To ensure the accurate annotation of the obstacle images during the tunnel pipe jacking construction, this study adopted the method of manual annotation. During the annotation process, the types of obstacles in each image were classified according to the actual situation and labeled using bounding boxes. The specific types of obstacles marked include broken and weak surrounding rocks, high-stress rock strata, expansive surrounding rocks, water and mud gushing, dangerous gases, rockburst and surrounding rock deformation, other obstacles and surface settlement and collapse. After marking, there were 425 broken and weak surrounding rocks, 366 high-stress rock strata, 512 expansive surrounding rocks, 332 water and mud gushing, 254 dangerous gas gushing, 163 rockburst and surrounding rock deformation, 295 other obstacles, and 320 surface settlement and collapse. The annotation process was carried out by two independent annotators, and Cohen's Kappa was adopted to measure the consistency among the annotators. The results show that the Cohen's Kappa value between the two annotators was 0.92, indicating high consistency and that the annotation results were reliable and accurate. To enhance the authenticity of the test results and adaptability to the actual construction environment, the study's test environment simulated various condition changes that may occur during tunnel pipe jacking construction to ensure stable model application in harsh construction environments. To verify the accuracy and classification ability of the model, the confusion matrix and Precision-Recall (PR) curve of the correct recognition times of the GCD-YOLOv5 model were analyzed, as shown in Figure 9.




(a) Confusion matrix of each type of obstacle identification (b) Experimental results of PR curve

Figure 9: Confusion matrix and PR curve of the number of correct obstacle identifications

Based on the updated confusion matrix calculations, the model performs well in most categories, especially in Soft Rock (SR) and Expansive Surrounding Rock (ER), with accuracy rates of 96.6% and 95.9%, respectively, demonstrating strong classification ability in these categories. The accuracy for Surrounding Rock (WR) and Settlement Collapse (SC) was 94.1% and 93.8%, indicating stable and reliable classification performance in these categories as well. However, the accuracy for Rock Bursts and Surrounding Rock Deformation (BD) and High-Stress Rock Layers (HG) was relatively lower, at 87.0% and 83.5%, possibly due to the complexity or lower sample numbers of these categories, which leads to slightly weaker performance. Other Obstacles (OH) achieved an accuracy of 90.7%, showing stable classification results. Figure 9(b) shows the PR curves for different obstacle types. The area under the PR curve was the smallest for fractured weak surrounding rock at 0.834, while it was the largest for other obstacles at 0.975, demonstrating the model's robust classification capabilities. These results confirm that the proposed lightweight GCD-YOLOv5 model achieved high accuracy and robust classification performance in real-world applications. To further verify whether the proposed model outperformed other models in recognition accuracy, the study analyzed the number of correctly identified images for each model, as shown in Figure 10.

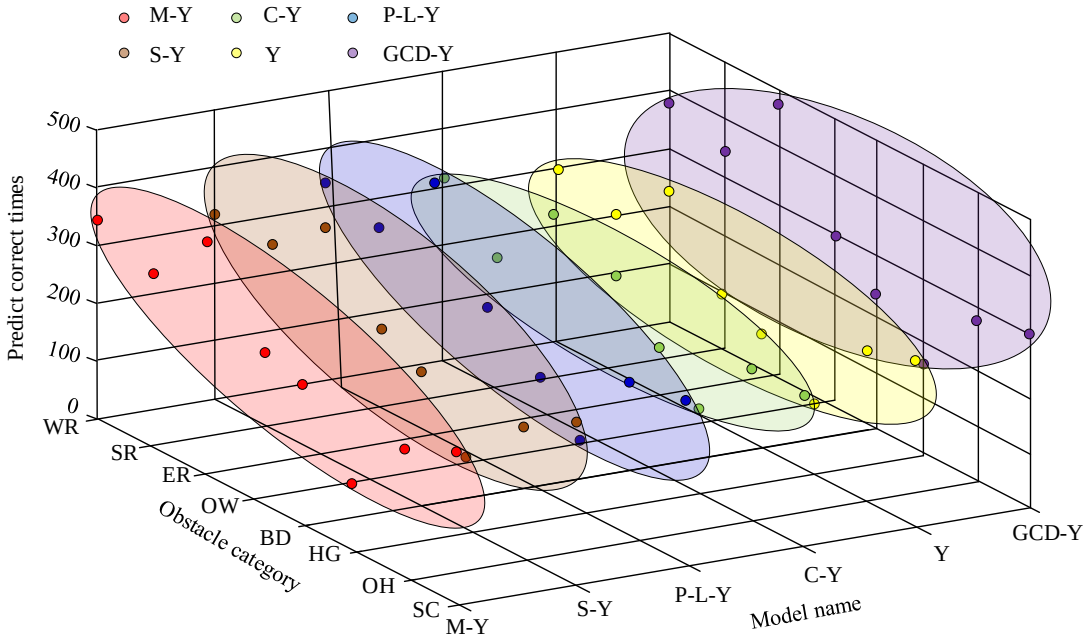



Figure 10: Statistics of the number of correct obstacle identifications

As shown in Figure 10, the GCD-YOLOv5 model correctly identified 2,498 images, achieving an accuracy of 94.0%. In comparison, the M-YOLOv5, S-YOLOv5, P-L-YOLOv5, and C-YOLOv5 models correctly identified 2,360, 2,336, 2,459, and 2,408 images, with accuracies of 88.5%, 87.6%, 92.2%, and 90.3%, respectively. All comparison models had lower accuracy than the proposed model. The original YOLOv5 model, without any

optimization, correctly identified only 2,150 images, achieving an accuracy of 80.6%, which was significantly lower than that of the proposed model. These results confirmed that the proposed model achieved higher accuracy than the other models in real-world applications.

To verify whether the model's efficiency was improved in practical applications, the study analyzed the Image Recognition Speed (FPS) of each model when processing obstacle images, as shown in Figure 11.

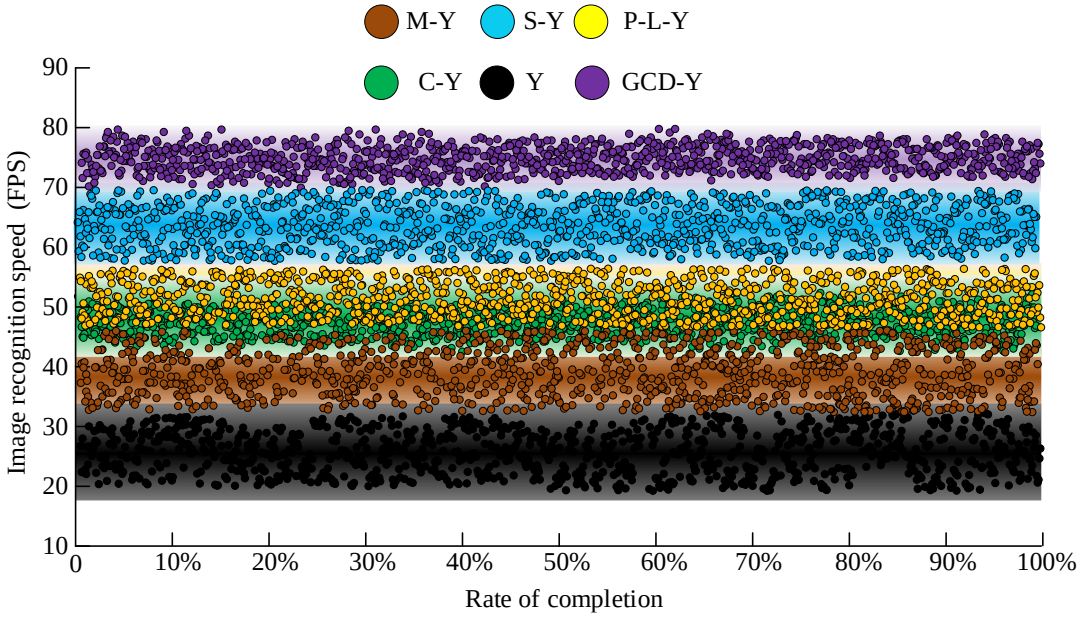



Figure 11: Comparison of image recognition speed and completion rate

As shown in Figure 11, when each model processes the obstacle images in actual construction, the FPS remains relatively stable. However, compared with the original YOLOv5 model, the FPS distribution of the proposed GCD-YOLOv5 model was narrower, indicating that it was more stable during operation. Specifically, the average FPS of the proposed model was 75, significantly higher than those of the other models. In contrast, the average FPS of the original YOLOv5 model when processing obstacle images was approximately 25 FPS, with the processing speed increasing by about 200%. To further verify the real-time performance of the model, the study decomposed the processing delay in detail. Among them, the image acquisition link took approximately 5ms, the preprocessing operation (such as image scaling and standardization) took approximately 10ms, the reasoning process took 15ms, and the post-processing (including non-maximum suppression) required approximately 4ms. Overall, the total processing delay was 34 ms, corresponding to 75 FPS, which proves that the proposed GCD-YOLOv5 model has high stability and real-time performance in actual scenarios. To comprehensively evaluate the detection performance of the GCD-YOLOv5 model, this study compares it with four other models: M-YOLOv5, S-YOLOv5, PL-YOLOv5, and C-YOLOv5, using the same test dataset. The evaluation metrics include mAP@0.5, mAP@0.5:0.95, Precision, Recall, and F1-score. The comparison results are shown in Table 3.

Table 3: Detection performance comparison of different models

| Model | mAP@0.5 (%) | mAP@0.5:0.95 (%) | Precision (%) | Recall (%) | F1-score (%) |
|---|---|---|---|---|---|
| M-YOLOv5 | 90.37 | 71.12 | 89.21 | 85.46 | 87.31 |
| S-YOLOv5 | 88.64 | 69.78 | 87.86 | 84.02 | 85.88 |
| PL-YOLOv5 | 91.43 | 73.36 | 90.68 | 86.81 | 88.67 |
| C-YOLOv5 | 89.91 | 72.04 | 89.63 | 85.57 | 87.42 |
| GCD-YOLOv5 | 93.62 | 76.24 | 92.41 | 89.38 | 90.42 |

As shown in Table 3, the GCD-YOLOv5 model achieved 93.62% in mAP@0.5 and 76.24% in mAP@0.5:0.95, outperforming all other models. For precision, recall, and F1-score, GCD-YOLOv5 also ranks the highest with values of 92.41%, 89.38%, and 90.42%, respectively. In comparison, the best-performing alternative model, PL-YOLOv5, achieved 91.43% in mAP@0.5, 73.36% in mAP@0.5:0.95, 90.68% in precision, 86.81% in recall, and 88.67% in F1-score. The other models show relatively lower results across most metrics. In addition, one-way ANOVA and paired t-tests were conducted on mAP@0.5 and F1-score across the five models. The statistical analysis indicates that GCD-YOLOv5 shows significant differences compared to the other models in both metrics ($p < 0.01$), confirming its advantage in overall detection performance.

## 5 Discussion

In this study, the performance of the GCD-YOLOv5

model was compared with C-YOLOv5, M-YOLOv5, and P-L-YOLOv5 on real-world tunnel images. The experimental results show that GCD-YOLOv5 outperforms the other models in both accuracy and processing speed. GCD-YOLOv5 achieved an accuracy of 94.0%, which was approximately 5.5%, 6.3%, and 3.5% higher than C-YOLOv5, M-YOLOv5, and P-L-YOLOv5, respectively. Furthermore, GCD-YOLOv5 operates at 75 FPS, significantly higher than the original YOLOv5 and other optimized models, demonstrating superior real-time processing capability. The superior performance of GCD-YOLOv5 can be attributed to its optimized architecture. We replaced the CSP network in YOLOv5 with Ghost Convolution and Depthwise Convolution networks. This architectural optimization effectively reduces computational load and parameter count while retaining efficient feature extraction capabilities, thereby improving both accuracy and speed. Additionally, GCD-YOLOv5 combines pruning with knowledge distillation during the training process, effectively mitigating the slight accuracy loss caused by pruning and ultimately achieving higher accuracy and smaller memory usage. In terms of model optimization, GCD-YOLOv5 adopts the a-IOU loss function and BN alignment techniques. These optimizations improve the model's stability and ensure that, even with significant pruning, the model can maintain high detection accuracy. Compared to other models, these innovative optimizations make GCD-YOLOv5 more suitable for real-world applications such as tunnel jacking, where rapid response and accurate obstacle detection are essential.

However, the advantages of GCD-YOLOv5 come with trade-offs. Although pruning and knowledge distillation optimize the model's computational efficiency, pruning in the early stages of training can lead to a slight drop in accuracy. Particularly in the initial epochs, large-scale pruning results in some accuracy loss. However, this loss was recovered through subsequent training and knowledge distillation, allowing the model to achieve a balance between accuracy and efficiency in the end. In conclusion, GCD-YOLOv5 achieved high accuracy while optimizing computational efficiency through architectural improvements, pruning, and knowledge distillation. While pruning may result in a slight decrease in accuracy in the early stages, it was eventually compensated for, making GCD-YOLOv5 highly advantageous in practical applications such as tunnel jacking construction, where real-time performance and accuracy were crucial, especially in resource-constrained environments.

# 6 Limitations and future research

Although the GCD-YOLOv5 model performs well on the GTX 1080Ti platform, it has not yet been tested on mobile-class GPUs or edge devices (such as Jetson boards), so the model's performance on resource-constrained devices has not been fully validated. Future research will focus on performance testing on mobile GPUs and Jetson platforms to ensure the model's adaptability and stability. Additionally, future work could explore hardware acceleration optimizations (such as TensorRT, OpenVINO) to improve inference speed on edge devices and consider transfer learning across different hardware platforms to enhance the model's cross-platform generalization. Furthermore, this study did not incorporate visualizations such as Grad-CAM due to the limitations of the current experimental environment, which does not support the extraction and processing of intermediate feature layers. Future work will consider integrating adaptable visualization techniques on more flexible testing platforms to enhance the model's interpretability and result clarity. Lastly, the model relies on well-lit and clear images, and environmental factors such as low lighting or complex backgrounds may affect detection accuracy. When facing new types of obstacles, additional training and optimization may be required. Therefore, future research could introduce a more diverse dataset to improve the model's robustness and explore transfer learning-based solutions to better generalize the model to unknown obstacle types.

# 7 Conclusion

To address the issues of model complexity and long processing time in obstacle recognition for tunnel pipe jacking construction, this study developed a lightweight model based on GCD-YOLOv5. The model replaced the CSP network with the GCD network and optimized output results using the a-IOU function. Additionally, pruning and knowledge distillation were applied to achieve model lightweighting. The test results showed that the proposed model achieved significantly higher accuracy than the unoptimized YOLOv5 model. The model had 0.8M parameters, 2.5G floating-point operations, a size of 1.8MB, and a response time of 19ms per image, all of which outperformed the compared models. In the recognition of actual tunnel pipe jacking construction obstacles, the model achieved an accuracy of 94.0% and a processing speed of 75 images per second, enabling fast and accurate obstacle classification in practical applications. In summary, the GCD-YOLOv5-based obstacle recognition model effectively met the lightweight requirements of harsh construction site conditions while maintaining considerable accuracy.

# Fundings

The research is supported by: Power China Roadbridge Group Co., Ltd, Research on key technologies for construction safety management throughout the entire process of pipe jacking operation. (Scientific and technological breakthroughs project: ZGDJLQ-YX-2023-01-0001).

# References

[1] Rayner-Philipson M, Sheil B, Zhang P. Prediction of pipe-jacking forces using a physics-constrained

neural network. Machine Learning and Data Science in Geotechnics, 2025, 1(1): 24-34. https://doi.org/10.1108/MLAG-06-2024-0004

[2] Jiahuan Wang, Haixiao Jia, Xuejiao Bai, et al. Research on the location of railway train in tunnel based on factor graph optimization. Applied Computer Letters, 2023, 7(1)

[3] Xu Z, Chen B, Zhan X, Xiu Y, Suzuki C, Shimada K. A vision-based autonomous UAV inspection framework for unknown tunnel construction sites with dynamic obstacles. IEEE Robotics and Automation Letters, 2023, 8(8): 4983-4990. https://doi.org/10.1109/LRA.2023.3290415

[4] ZHAO X, DENG K, ZHANG Y, MA Y, XIA Y. Molding quality inspection method for large-diameter shield tunnels. Chinese Journal of Engineering, 2024, 46(2): 365-375.

[5] Puspita E, Suryadi D, Rosjanuardi R. Learning obstacles of prospective mathematics teachers: A case study on the topic of implicit derivatives. Kreano, Jurnal Matematika Kreatif-Inovatif, 2023, 14(1): 174-189.

[6] Jiahuan Wang, Haixiao Jia, Peifen Pan, et al. Research on the technology of man-machine collision early warning system in tunnels based on bds high-precision positioning in tunnel. Applied Computer Letters, 2023, 7(1)

[7] Yusro M M, Ali R, Hitam M S. Comparison of faster r-cnn and yolov5 for overlapping objects recognition. Baghdad Science Journal, 2023, 20(3): 0893-0893. https://doi.org/10.21123/bsj.2022.7243

[8] Gao P, Lee K, Kuswidiyanto L W, Hu K., Liang G, Han X. Dynamic beehive detection and tracking system based on YOLO V5 and unmanned aerial vehicle. Journal of Biosystems Engineering, 2022, 47(4): 510-520. https://doi.org/10.1007/s42853-022-00166-6

[9] Xiao D, Xie F, Gao Y, Xie H. A detection method of spangle defects on zinc-coated steel surfaces based on improved YOLO-v5. The International Journal of Advanced Manufacturing Technology, 2023, 128(1-2): 937-951. https://doi.org/10.1007/s00170-023-11963-4

[10] Zheng J C, Sun S D, Zhao S J. Fast ship detection based on lightweight YOLOv5 network. IET Image Processing, 2022, 16(6): 1585-1593. https://doi.org/10.1049/ipr2.12432

[11] Chen S, Liao Y, Lin F, Huang B. An improved lightweight YOLOv5 algorithm for detecting strawberry diseases. IEEE Access, 2023, 11: 54080-54092. https://doi.org/10.1109/ACCESS.2023.3282309

[12] Zhou M, Wu L, Liu S, Li J. UAV forest fire detection based on lightweight YOLOv5 model. Multimedia Tools and Applications, 2024, 83(22): 61777-61788. https://doi.org/10.1007/s11042-023-15770-7

[13] Xu Z, Chen B, Zhan X, Xiu Y, Suzuki C, Shimada K. A vision-based autonomous UAV inspection framework for unknown tunnel construction sites with dynamic obstacles. IEEE Robotics and Automation Letters, 2023, 8(8): 4983-4990. https://doi.org/10.1109/LRA.2023.3290415

[14] Yongcan C, Jiajie C, Haoran W, Yue F, Zhaowei L, Hui X. Key technology of underwater inspection robot system for large diameter and long headrace tunnel. Journal of Tsinghua University (Science and Technology), 2023, 63(7): 1015-1031.

[15] Naranjo J E, Valle A, Cruz A, Martín M, Anguera M, García P, Jiménez F. Automation of haulers for debris removal in tunnel construction. Computer‐Aided Civil and Infrastructure Engineering, 2023, 38(14): 2030-2045. https://doi.org/10.1111/mice.12997

[16] Li Y, Bao T, Li T, Wang R. A robust real‐time method for identifying hydraulic tunnel structural defects using deep learning and computer vision. Computer‐Aided Civil and Infrastructure Engineering, 2023, 38(10): 1381-1399. https://doi.org/10.1111/mice.12949

[17] Zhang, W, Zhang W, Zhang G, Huang J, Li M, Wang X, Guan X. Hard-rock tunnel lithology identification using multi-scale dilated convolutional attention network based on tunnel face images.Frontiers of Structural and Civil Engineering, 2023, 17(12), 1796-1812. https://doi.org/10.1007/s11709-023-0002-1

[18] Alsuwaylimi A A, Alanazi R, Alanazi S M,Alenezi S M, Saidani T, Ghodhbani R. Improved and efficient object detection algorithm based on yolov5. Engineering, Technology & Applied Science Research, 2024, 14(3): 14380-14386. https://doi.org/10.48084/etasr.7386

[19] Jooshin H K, Nangir M, Seyedarabi H. Inception‐YOLO: Computational cost and accuracy improvement of the YOLOv5 model based on employing modified CSP, SPPF, and inception modules. IET Image Processing, 2024, 18(8): 1985-1999. https://doi.org/10.1049/ipr2.13077

[20] Jing N, Hu Y, Wang Y. Research on Sign Language Recognition for Hearing-Impaired People Through the Improved YOLOv5 Algorithm Combining CBAM with Focal CioU. Informatica, 2025, 49(14). https://doi.org/10.31449/inf.v49i14.7596

[21] Bakr E, Alsaedy Y, Elhoseiny M. Look around and refer: 2d synthetic semantics knowledge distillation for 3d visual grounding. Advances in neural information processing systems, 2022, 35: 37146-37158.

[22] Johnson M B, Hyman S E. A critical perspective on the synaptic pruning hypothesis of schizophrenia pathogenesis. Biological Psychiatry, 2022, 92(6): 440-442. https://doi.org/10.1016/j.biopsych.2021.12.014

[23] Yu K, Tang G, Chen W, Hu S, Li Y, Gong H. MobileNet-YOLO v5s: An improved lightweight method for real-time detection of sugarcane stem nodes in complex natural environments. Ieee Access, 2023, 11(1): 104070-104083. https://doi.org/10.1109/ACCESS.2023.3317951

[24] Zhang S, Yang H, Yang C, Yuan W, Li X, Wang X.

Edge device detection of tea leaves with one bud and two leaves based on ShuffleNetv2-YOLOv5-Lite-E. Agronomy, 2023, 13(2): 577-578. https://doi.org/10.3390/agronomy13020577

[25] Xue J, Chen H, Hu Y, Chen M, Wu L I, Chang X. Reduce Detection Latency of YOLOv5 to Prevent Real-Time Tracking Failures for Lightweight Robots, Proceedings of the 15th Asia-Pacific Symposium on Internetware. 2024: 437-446. https://doi.org/10.1145/3671016.3671392

[26] Gao X, Zhang L, Chen X, Lin C, Hao R, Zheng J. GCT-YOLOv5: a lightweight and efficient object detection model of real-time side-scan sonar image. Signal, Image and Video Processing, 2024, 18(1): 565-574 https://doi.org/10.1007/s11760-024-03174-5.