Computers, Materials & Continua

Tech Science Press

Doi:10.32604/cmc.2025.065251

ARTICLE

FSS-YOLO: The Lightweight Drill Pipe Detection Method Based on YOLOv8n-obb

Mingyang Zhao[1,2,*], Xiaojun Li[1,3], Miao Li[1,2] and Bangbang Mu[1,2]

[1]School of Energy Science and Engineering, Henan Polytechnic University, Jiaozuo, 454003, China
[2]School of Innovation and Entrepreneurship, Henan Polytechnic University, Jiaozuo, 454003, China
[3]Henan International Joint Laboratory of Coalmine Ground Control, Jiaozuo, 454003, China

*Corresponding Author: Mingyang Zhao. Email: zhao1999116172@163.com

Received: 07 March 2025; Accepted: 08 May 2025; Published: 03 July 2025

ABSTRACT: The control of gas extraction in coal mines relies on the effectiveness of gas extraction. The main method of gas extraction is to drive drill pipes into the coal seam through a drilling rig and use technologies such as hydraulic fracturing to pre-extract gas in the drill holes. Therefore, the real-time detection of the drill pipe status is closely related to the effectiveness of gas extraction. To achieve fast and accurate identification of drill pipes, we propose FSS-YOLO, which is a lightweight drill pipe detection method based on YOLOv8n-obb. This method first introduces the FasterBlock module into the C2f module of YOLOv8n-obb to reduce the number of model parameters and decrease the computational cost of the model and redundant feature maps. Next, the SimAM attention mechanism is added to the backbone network to enhance the weight of important features in the feature map and improve the model's feature extraction capability. In addition, using shared convolution to optimize the detection head, not only lightens the detection head but also enhances its ability to learn features of different scales, improving the model's generalization ability. Finally, the FSS-YOLO algorithm is validated on the self-built dataset. The results show that compared with the original algorithm, FSS-YOLO achieves improvements of 5.1% in mAP50 and 11.5% in Recall, reduces the number of parameters by 45.8%, and achieves an inference speed of 27.8 ms/frame on Jetson Orin NX. Additionally, the visual detection results for different scenarios demonstrate that the improved YOLOv8n-obb algorithm has promising application prospects.

KEYWORDS: Gas extraction; YOLOv8n-obb; SimAM; shared Conv; coal mine; intelligent coal mine

1 Introduction

The hydraulic punching pressure relief and permeability enhancement technology of coal seams is one of the main methods for coal reservoir transformation and enhancing gas extraction [1,2]. The drilling used for hydraulic punching is completed by drilling drill pipes into the coal and rock layers. However, the underground workers do not always operate the drilling rig according to the standard procedures. For example, during the drilling process, work is stopped before reaching the specified drilling depth, and the number of drill pipes is falsely reported during the withdrawal process to quickly earn work fees. This poses great challenges and safety hazards to coal mine gas prevention. In addition, there are many problems with arranging manual monitoring of changes in the workflow of drill pipes, such as low efficiency, high labor costs, and deviation caused by human visual fatigue, which makes it difficult to ensure the effectiveness of gas extraction. In recent years, with the development of coal mine intelligent mining system theory and intelligent mine system engineering in coal mines [3,4], computer vision has been significantly applied in

the coal mining industry and has achieved excellent results [5–8]. Therefore, developing an intelligent drill pipe status detection method based on computer vision is of great significance.

As shown in Fig. 1, the drilling rig has different inclinations during operation, and using a rotating object detection algorithm is more suitable for detecting drill pipes. However, domestic and foreign scholars lack application studies on detecting rotating drill pipes, and the complex underground environment in coal mines (such as uneven lighting, low illumination, heavy dust and fog, and variable target size) can lead to a decline in algorithm performance. Using the default YOLOv8n-obb detection algorithm may not achieve high-precision object recognition. Furthermore, coal mines currently tend to use AI terminal devices that are easier to deploy, lower in cost, and have limited computing power. This requires us to consider the model's operating speed while improving the algorithm's accuracy.

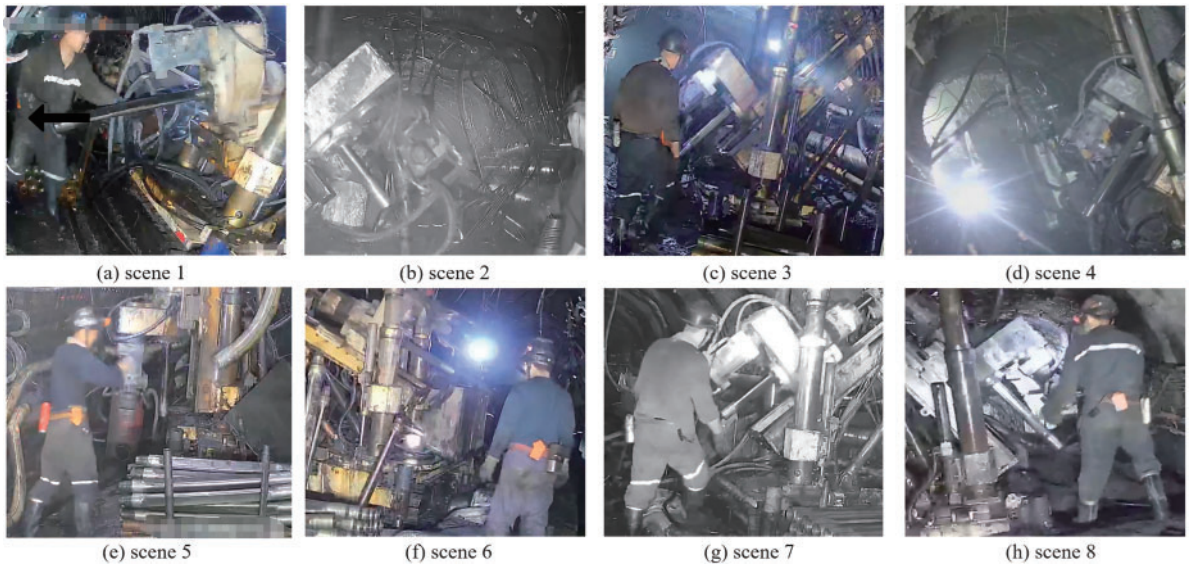


Figure 1: Working scenarios with different rotation angles, lighting conditions, dust and fog conditions, varying distances and infrared photography

In summary, this paper proposes FSS-YOLO, a high-accuracy and fast detection network based on YOLOv8n-obb, designed to identify drill pipes with angles. This algorithm still exhibits good recognition performance and inference speed in different underground scenarios. The main contributions of this paper are as follows:

(1) We collected a large number of videos of drilling rig operations in different scenarios and created a drill pipe dataset based on the videos to train and evaluate the performance of different detection algorithms.
(2) We analyzed the image redundancy issues during the creation of the drill pipe dataset and used C2f-Faster to reduce the computational cost of redundant feature maps and decrease the model's parameter count.
(3) We introduced the SimAM attention mechanism into the backbone network, enhancing the model's focus on important features without introducing additional parameters.
(4) We optimized the default detection head using shared convolution and replaced all the BN layers in the detection head with IN layers, improving the model's generalization ability while reducing the model's parameter count.

2 Related Work

The object detection algorithm based on deep learning [9] has been successfully applied in multiple fields of the coal mining industry. Among them, the YOLO [10–12] series algorithms are particularly widely used in coal mines due to their efficiency and accuracy. Domestic and foreign scholars mainly apply the YOLO algorithm to coal mines in two ways: one is to directly use YOLO for object detection, such as coal gangue detection [13–15], conveyor belt foreign object detection [16,17], safety helmet detection [18–20], and underground personnel position detection [21–23]; Another approach is to combine it with other technologies to expand its functionality, such as integrating with tracking algorithms to achieve real-time monitoring and tracking of miners [24], combining with segmentation networks to achieve coal-rock interface recognition [25]. In the application of drill pipe detection, there are three main approaches: First, using image classification algorithms to distinguish different working states of the drilling rig [26] and to classify different actions of workers loading and unloading drill pipes [27]. Second, using object detection algorithms to locate the positions of the drilling rig, drill bit, drill pipe, and workers [28]. Third, using semantic segmentation algorithms to calculate the mask area of the drill pipe [29].

In terms of lightweight improvement, Wu et al. replaced the conventional convolution in the backbone network of YOLOv5s with Ghost Conv in the real-time monitoring method of lump coal on conveyor belts, and built an AMGC-YOLO network, successfully reducing the number of model parameters [30]. However, they also introduced the CBAM attention mechanism after Bottleneck, increasing the number of model parameters, and the improvement in mAP50 was minimal. Ruan Shunling et al. introduced depthwise separable convolution and SE attention mechanism into the C2f of YOLOv8n and PConv into the detection head in the mining area obstacle detection model, constructing a BPI-YOLO lightweight network [31]. Although it reduced redundant calculations and the number of model parameters, achieving the goal of lightweighting the model, the mAP50 of the model also decreased slightly. Yue Zhongwen et al. used the MobileNetv3-Small and CBAM attention mechanism to construct a backbone network for a lightweight blast-hole intelligent detection method [32], successfully reducing model parameters while ensuring accuracy. However, in hardware deployment testing, the FPS of the model was reduced. Du et al. [33] used ShuffleNetv2 as the backbone network of YOLOv8 and replaced the conventional convolution in ShuffleNetv2 Block and neck network with depthwise separable convolution. Finally, the SE attention mechanism was introduced in the ShuffleNetv2 Block to ensure model accuracy.

In summary, the current improvement measures for the YOLO algorithm in coal mining application scenarios tend to use DWConv for model lightweighting, as well as attention mechanisms such as CBAM, SE, and EMA to ensure model accuracy. However, compared to the two lightweight convolutions FasterBlock and DWConv, DWConv has the problem of excessive memory access, which can reduce the running speed of the model. The SimAM attention mechanism is a 3D weighted attention mechanism that does not introduce additional parameters, making it more suitable for ensuring the accuracy of a lightweight model. Moreover, among the current improvement measures, no research has taken into account the use of shared convolution. Of course, this is limited by the fact that the recognition objects in coal mine application scenarios are relatively single, and only a few scenarios require the recognition of different objects of three scales: large, medium, and small. Therefore, we adopted a combination of FasterBlock, SimAM, and shared convolution in view of the particularity of the task in this paper.

3 Dataset

In the self-built dataset, we fully considered the scene factors such as shooting angles, lighting conditions, shooting distances, dust and fog conditions, the working angles of the drilling rigs, different occlusion situations, and infrared shooting. Moreover, the data collection spans from November 2023 to July 2024. The

underground monitoring equipment in coal mines moves as the working face advances, combined with the special lighting conditions and dust and fog phenomena in the roadway, resulting in diverse backgrounds of underground monitoring images and increasing the difficulty of YOLO model generalization. Therefore, to collect images in different scenarios more comprehensively, this paper's dataset collected monitoring videos of 19 drilling sites and 176 drilling holes. By selecting and deduplicating all the videos, a total of 90 videos from different scenes were obtained. Based on the duration of the videos from different scenes, one frame was extracted every 3–5 s, resulting in 8325 images. The dataset was divided into 6765 images for the training set and 1560 images for the validation set, with all images having a size of 1920 × 1080 pixels.

In addition, during the process of creating the dataset, it was found that there was a high similarity between the images of drilling withdrawal. This is not a good thing for the dataset, as it means that the model will generate redundant feature maps during feature extraction. Therefore, further analysis of the drilling withdrawal dataset reveals that the root cause comes from its workflow. The workflow of drilling withdrawal is to withdraw the drill pipe from the borehole through the reciprocating motion of the power head until all the drill pipes are removed from the borehole. The changing factors in this process include the actions and positions of workers, lighting conditions, drill pipe length, and power head position. However, the withdrawal process of all drill pipes is the same. If the withdrawal process of each drill pipe is taken as one cycle, the changing factors in a single drilling scene will have periodic changes, which will lead to the problem of high similarity between images. This is illustrated in Fig. 2.

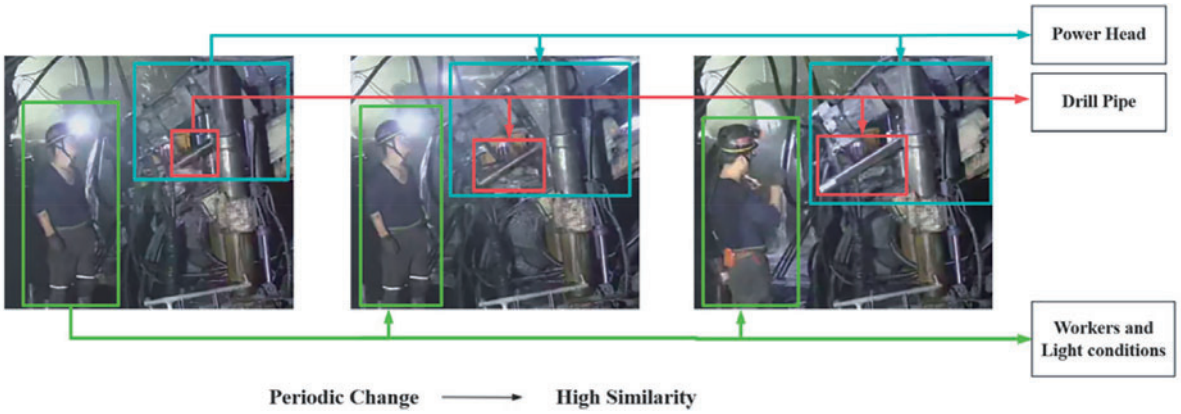


Figure 2: During the reciprocating motion of the power head, the drill pipe is slowly withdrawn from the borehole, and the length of the drill pipe in the image increases accordingly. As the worker removes the drill pipe and places it in the designated area, there will be changes in its position and lighting. Since the withdrawal process of each drill pipe is the same and exhibits periodic changes, the frames extracted from the video will also show high similarity

Regarding the limitations of the dataset:

(1) Although 90 different scenarios have been collected in the self-built dataset, compared with the complex and changeable working environment of the roadway, the number of scenarios still needs to be increased. Moreover, the total number of images in the dataset is only 8325, which is far lower than that of the ImageNet or COCO datasets.

(2) The self-built dataset is obtained by dynamically extracting frames according to the length of the collected scene videos (extracting one frame every 3–5 s). This may limit the sample distribution to some extent. Therefore, the number of images in each scenario is not equal (the number of images in

each scenario ranges from 50 to 250). Due to the limitations of the workflow, some images have high similarity, which may result in a small number of redundant images.

4 Lightweight Drill Pipe Detection Algorithm

4.1 Overall Technical Roadmap

As shown in Fig. 3. Firstly, use a camera to record the working video of the drilling site, extract frames from the video, clean the data, and divide it into training sets and validation sets. Then, use a 4090 server to train and optimize the network structure, and obtain the FSS-YOLO algorithm proposed in this paper. Secondly, FSS-YOLO is deployed on AI terminal devices (Jetson Orin NX in this paper) to detect real-time video streams transmitted underground and output the detection results. Finally, the visualization results are saved to the database of the coal mine and displayed at the intelligent monitoring center of the coal mine.

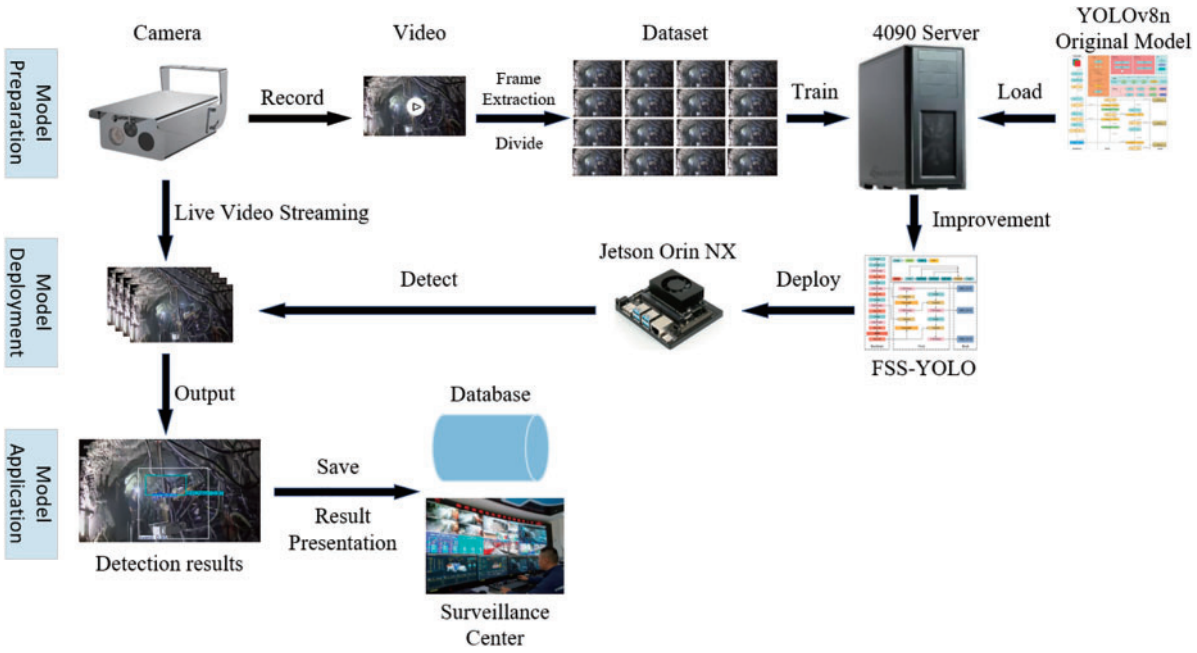


Figure 3: Overall flowchart

4.2 FSS-YOLO

As shown in Fig. 4, to improve the accuracy and speed of the YOLOv8n-obb algorithm in identifying drill pipes, the FasterBlock from FasterNet is first used to replace the Bottleneck in C2f, resulting in the C2f-Faster module. The C2f in the backbone and neck networks is then replaced with C2f-Faster to reduce the computational cost of the model and redundant feature maps. Next, the SimAM attention mechanism is introduced after each C2f-Faster in the backbone network to enhance the important features in the C2f-Faster output feature map. To further improve feature fusion, the SimAM attention mechanism is also added after the SPPF layer to increase the weight of important features and enhance the model's attention to important features without introducing additional parameters. Finally, shared convolution is introduced into the default detection head, allowing the detection head to learn more general features instead of features from a single scale, improving the model's generalization ability. Additionally, the BN layers in the detection head are replaced with IN layers to further enhance the model's detection accuracy.

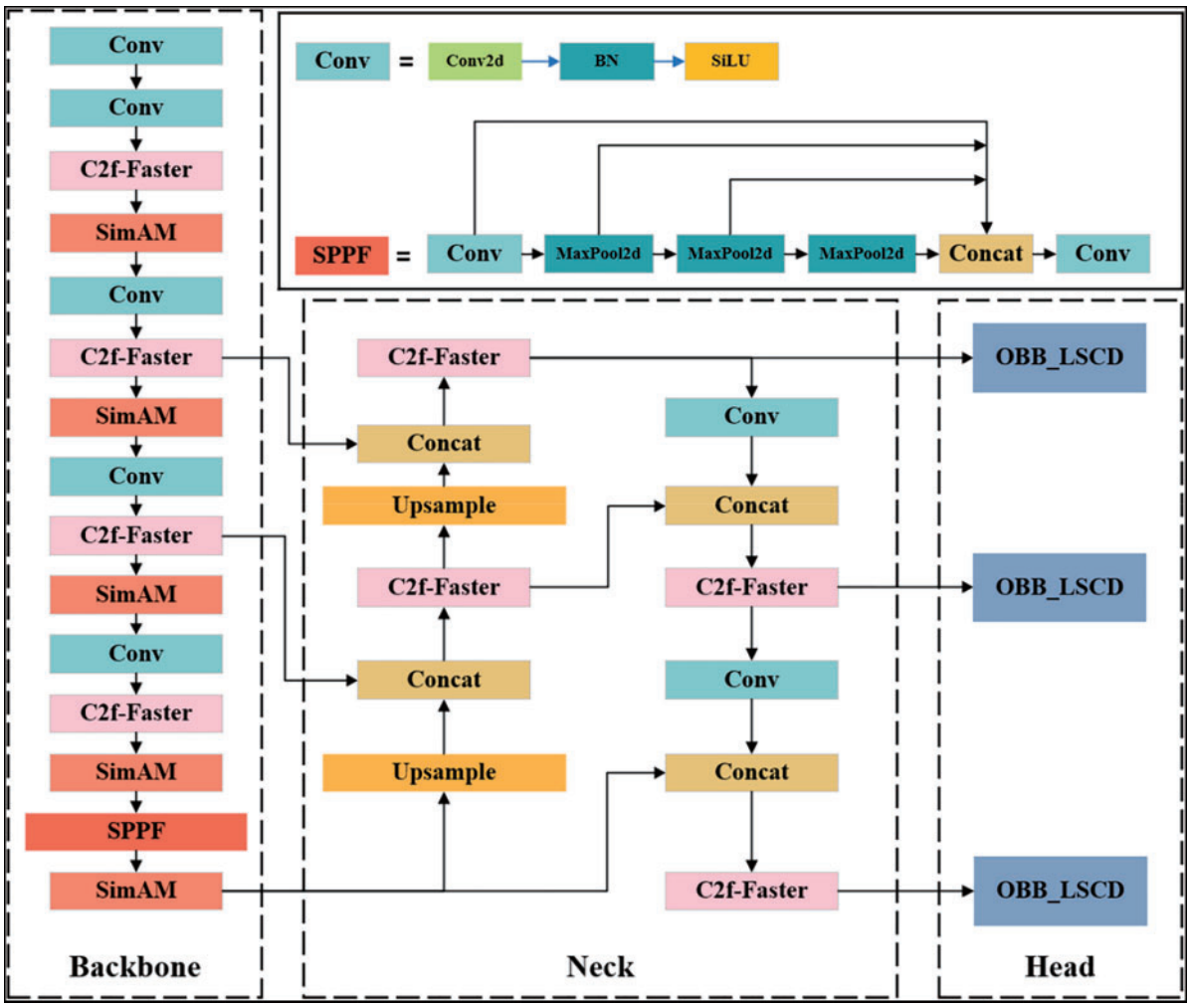


Figure 4: FSS-YOLO network architecture diagram

4.3 C2f-Faster

As described in Section 2, the high similarity of images in the same scene results in redundant feature maps. As shown in Fig. 5, this redundancy not only exists in different images in the same scene but also among multiple feature maps of the same image. Moreover, redundant feature maps do not bring learnable new features to the model, while also performing computations with minimal significance. This leads to only slight improvements in model accuracy and an increase in latency. As shown in Fig. 6, FasterNet [34] reexamined redundancy and memory access in feature maps and proposed a simple and efficient PConv that simply applies conventional convolution kernels to some channels of the input feature map for feature extraction, while keeping the remaining input feature map unchanged.

The FLOPs of PConv is:

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

This is lower than the FLOPs of conventional convolution:

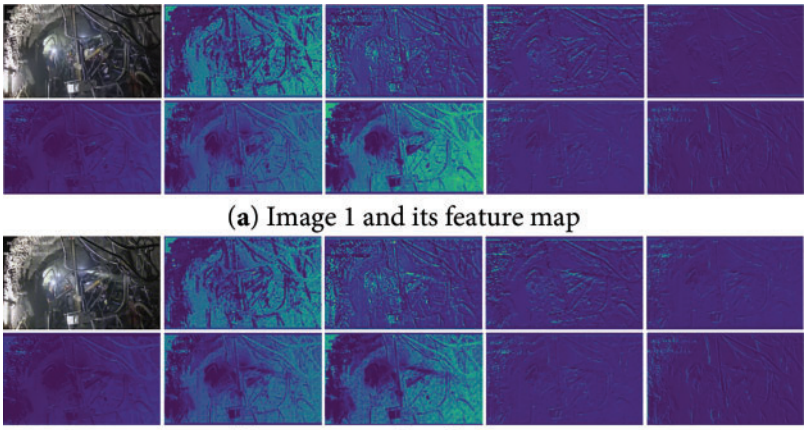
$$h \times w \times k^2 \times c^2 \tag{2}$$

When $r = c_p/c = 1/4$, FLOPs is only 1/16 of conventional convolution.

To compensate for the decrease in accuracy caused by the reduction of feature maps, FasterNet adds two pointwise convolution PWConv layers after the PConv, making full use of all channel information. The FLOPs of the combination of PConv and PWConv are calculated as:

$$h \times w \times \left(k^2 \times c_p^2 + c^2\right) \tag{3}$$

Although the stacking of the Bottleneck in C2f improves the model's ability to extract features, the stacked convolutional layers also generate redundant feature maps. Therefore, replacing Bottleneck with FasterBlock can reduce the generation of redundant feature maps and reduce the impact of redundant calculations on the adjustment of model parameters, which can help the model learn more important features. Moreover, As shown in Fig. 7, the combination of PConv and PWConv makes the receptive field of the FasterBlock similar to that of T-shaped convolution. Compared with the square receptive field of the conventional convolution in the Bottleneck, the FasterBlock can better capture the spatial features in the feature maps, thereby improving the detection accuracy of the model.



(**a**) Image 1 and its feature map

(**b**) Image 2 and its feature map

Figure 5: Images 1 and 2 are two different images in the same scene. **(a,b)** are the feature maps of the two images in the middle layer of the network. It can be seen that the feature maps of different channels in the same image have high similarity, and the feature maps of different images in the same scene also have high similarity

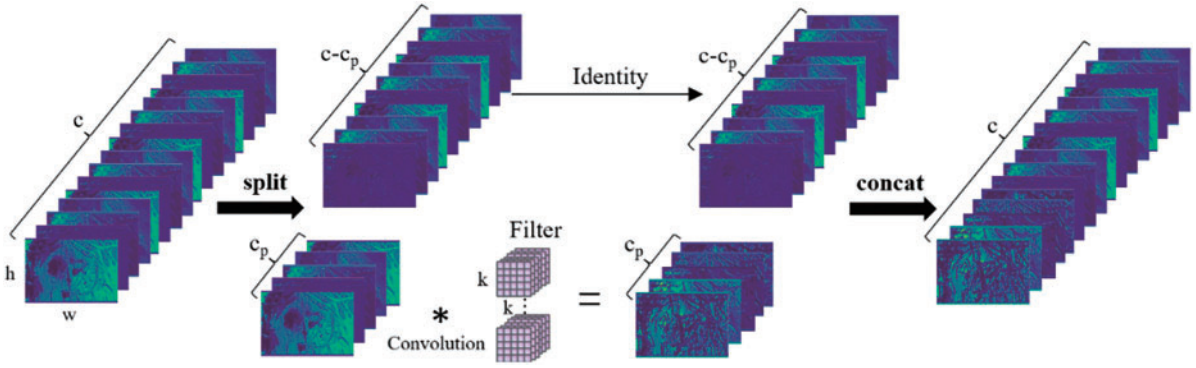


Figure 6: PConv schematic diagram

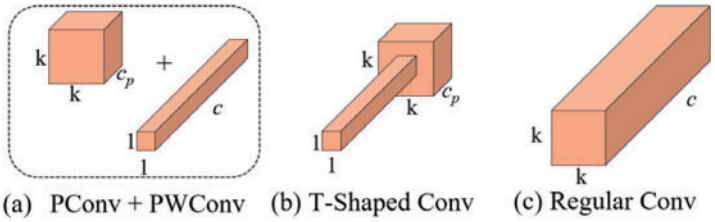


Figure 7: Different convolution structural diagram

C2f_Faster replaces the default Bottleneck module in C2f with FasterBlock, and its structure is shown in Fig. 8: Among them, taking c_p = c/4 and k = 3, PConv performs a 3 × 3 convolution on the first 1/4 of the input feature map's channels, applies an identity mapping to the remaining 3/4 of the channels, and then concatenates the two results. The FasterBlock module consists of PConv and PWConv. The feature map first goes through the PConv layer, which extracts features from some of the feature maps while reducing the computation of redundant feature maps. Then, it goes through the Conv layer with a size of 1 × 1 to increase the number of feature map channels and enhance the strength of feature extraction. Secondly, a 1 × 1 Conv2d layer is used to further extract features, align the input and output channel numbers, and perform DropPath regularization on the results to prevent model overfitting. Finally, connect the results with the Shortcut operation to complete feature fusion.

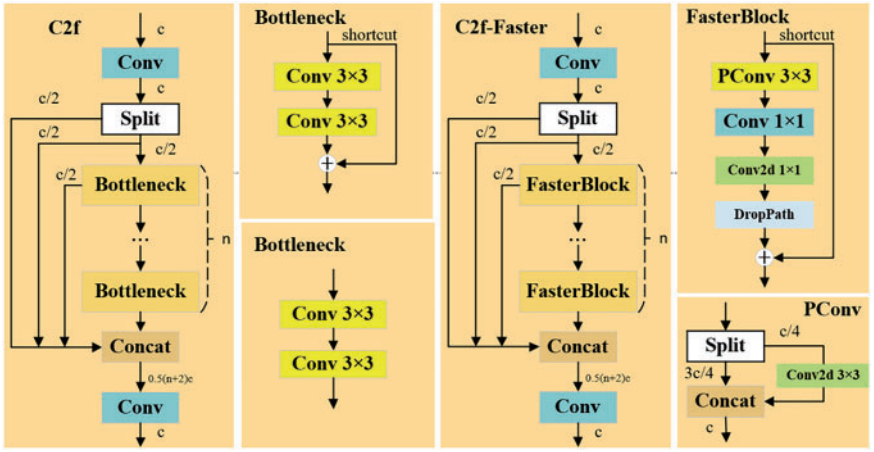


Figure 8: C2f-Faster structural diagram

4.4 SimAM Attention Mechanism

After replacing all C2f modules with C2f_Faster modules based on the original YOLOv8n-obb model, the number of model parameters was successfully reduced, and the accuracy of the model was slightly improved. However, the detection performance is still not ideal, with poor recognition results in some scenarios. Through the study of the existing YOLO network architecture, it has been found that introducing attention mechanisms can yield better results [35,36]. For example, SE [37] assigns different weights to different channels, allowing the model to adaptively learn the importance of each channel; CBAM [38] assigns different weights to different spatial locations, allowing the model to adaptively learn the importance of features at different locations, enhancing the model's ability to perceive features at various positions; as well as other attention mechanisms like CPCA [39] and EMA. However, considering the purpose of model

lightweighting, the attention mechanism introduced should not add extra parameters or excessively increase inference time. Therefore, this paper conducts experimental comparisons of several attention mechanisms, with the results shown in Section 5.4. Among them, the SimAM [40] attention mechanism is found to be suitable for the research needs of this paper.

The SimAM attention mechanism, based on CBAM and SE attention mechanisms, rethinks the relationship between channel features and spatial features. It assigns different weights to different spatial positions in different channels, achieving a 3D form of weighted attention mechanism. The implementation process does not introduce additional parameters, and compared to other attention mechanisms, the increase in computational cost is relatively small. Its principle is shown in Fig. 9.

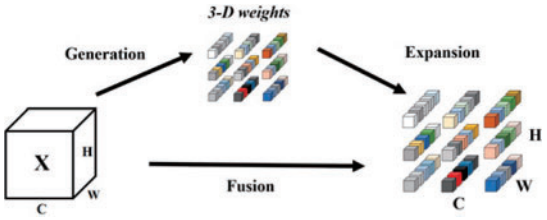


Figure 9: SimAM schematic diagram

SimAM is an attention mechanism based on the local self-similarity of feature maps, which assumes that information-rich pixels typically exhibit different influence patterns than surrounding pixels, and such pixels usually suppress the feature expression of surrounding pixels. Therefore, the SimAM calculates the average of the squared differences between each pixel and its neighboring pixels (after normalization) to indirectly reflect similarity, and the similarity between each pixel and its neighboring pixels can be represented as:

$$e_t^* = \frac{4\left(\hat{\sigma}^2 + \lambda\right)}{\left(t - \hat{\mu}\right)^2 + 2\hat{\sigma}^2 + 2\lambda} \tag{4}$$

where $\hat{\mu}$ and $\hat{\sigma}^2$ represent the mean and variance of all pixel points within the same feature map, respectively. The lower the similarity, the greater the difference between the pixel and its surrounding pixels, and the higher the visual importance. Therefore, the importance of each pixel can be obtained through $1/e_t^*$, and the pixel importance formula is:

$$\tilde{X} = sigmoid\left(\frac{1}{E}\right) \odot X \tag{5}$$

Here, E groups feature map in the channel and spatial dimensions, and the sigmoid function is used to constrain excessively large values in E. The SimAM attention mechanism dynamically adjusts the weight of each pixel based on the importance calculated by the Formula (5), achieving enhancement of important features and suppression of irrelevant features.

4.5 OBB-LSCD Lightweight Detection Head

The rotating object detection head OBB of YOLOv8n-obb uses three sets of convolution groups with the same structure to predict objects of different scales, as shown in Fig. 10. Although these convolution groups with the same structure can perform prediction tasks well, they are not conducive to the lightweight deployment of the model (the parameters of OBB account for about 26.6% of the total model parameters). Furthermore, the convolution kernels in each convolutional group can only learn feature maps at a single

scale, which is not beneficial for improving the model's generalization ability. Therefore, to achieve the goal of a lightweight model, this paper optimizes the default detection head using shared convolution, resulting in the OBB-LSCD (OBB Lightweight Shared Conv Detect) lightweight detection head.

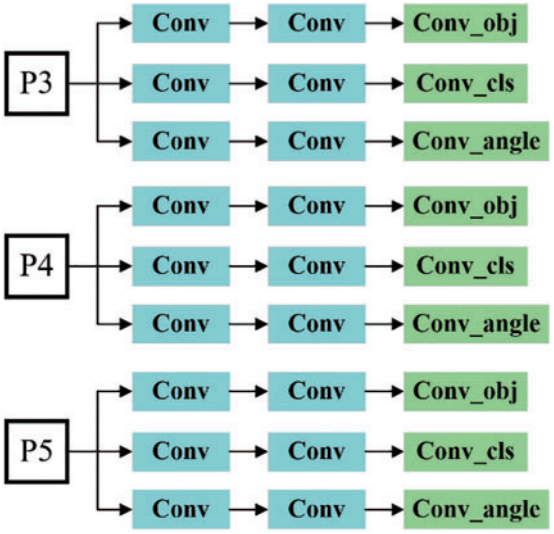


Figure 10: OBB structural diagram

As shown in the dashed box in Fig. 11, a set of shared convolutions is used to replace the convolution branches for object classification and localization in each convolutional group of OBB. The output feature maps of P3, P4, and P5 will be sequentially computed with this set of shared convolution, which can help the model better learn general features rather than relying on features at a fixed scale, making it more adaptable to untrained data and thus improving the model's generalization ability. Finally, different scales of scale layers are used to scale the features, to solve the issue that shared convolutions cannot handle the inconsistency in object scales during prediction. To address the problem of mismatch between the number of input channels and the number of output channels of P3, P4, and P5 in shared convolution, a Conv layer is added after P3, P4, and P5 to adjust the output channel numbers.

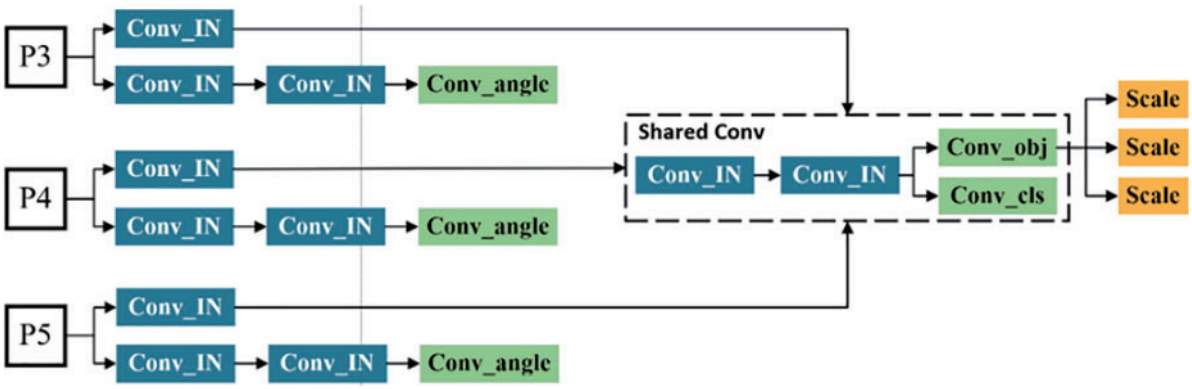


Figure 11: OBB-LSCD structural diagram

Shared convolution can significantly reduce the number of parameters while saving computational resources, as the same set of parameters can be reused multiple times during forward and backward propagation of the model, reducing computation and memory consumption. This is particularly suitable for AI devices with limited computing power.

The dataset in this paper contains images with high similarity, leading to redundancy in the feature maps. Besides using the C2f_Faster to reduce the impact of redundant feature maps on model accuracy, this paper also adopts another approach, which is to reduce the impact of redundant feature maps on model accuracy by adding a normalization layer in the Conv layers of OBB-LSCD. To further improve the accuracy of the model, this paper compares the impact of BN (Batch Normalization), GN (Group Normalization), and IN (Instance Normalization) on model accuracy through experiments.

5 Model Training and Result Analysis

5.1 Experiment Environment

To accelerate the speed of model training, this paper uses high-performance servers. But this is not the final deployment environment for the model. Some coal mines tend to use AI terminal devices with lower costs and smaller space occupations as algorithm carriers. Therefore, in order to better compare and test the real-time detection performance differences between various algorithms, we used Nvidia Orin NX terminal devices as the hardware devices for the deployment environment.

The training environment in the experiment used a high-performance server. The operating system of the server is Ubuntu 22.04.5 LTS. The CPU is Intel® Xeon® Platinum 8352V, the GPU is an NVIDIA GeForce RTX 4090 24 G, the PyTorch version is 2.0.0, the Python version is 3.8.16, and the CUDA version is 11.8. The deployment environment in the experiment used Jetson Orin NX. The operating system of the server is Ubuntu 22.04.5 LTS. The CPU is 8 core Arm® Cortex®-A78AE v8.2 64-bit CPU 2 MB L2 + 4 MB L3, the GPU is 1024-core NVIDIA Ampere architecture GPU with 32 Tensor Cores, the PyTorch version is 2.2.0, the Python version is 3.8.16, and the CUDA version is 12.2.

The input image size is 640 × 640, BatchSize is 32, Optimizer uses SGD, and the initial and final learning rates are both 0.01. The training is set for 300 epochs. Meanwhile, the number of early stop epochs is set to 100, the momentum is set to 0.937, the weight_decay is 0.0005, and the mosaic data augmentation is enabled.

5.2 Evaluation Metrics

To validate the performance of FSS-YOLO, Precision, Recall, mean Average Precision at 50% (mAP50), inference time, GFLOPs, and parameters (Params) are selected as evaluation metrics. Among them, mAP50, as a core metric, can intuitively and comprehensively reveal the accuracy level of the algorithm, so it is regarded as a key decisive factor for evaluating the performance of the object detection algorithm. Meanwhile, inference time is a key evaluation metric for measuring the model's capability in real-time detection scenarios. Precision is the ratio of correctly predicted results among all positive samples. Recall is the ratio of correctly predicted positive samples among all the positive samples. The calculations are as follows, respectively:

$$P = \frac{TP}{TP + FP} \tag{6}$$

$$R = \frac{TP}{TP + FN} \tag{7}$$

mAP50 is the mean of each category AP under the condition of an IoU threshold of 0.5. AP is the area enclosed by the P-R curve and the coordinate axis, and mAP50 is calculated as follows:

$$AP = \int_0^1 P(r)\, dr \tag{8}$$

$$mAP50 = \frac{1}{n} \sum_{i=1}^{n} AP(i) \tag{9}$$

Speed refers to the inference time, which is the time of the model to infer an image, measured in milliseconds (ms).

5.3 Ablation Experiment

To verify the effectiveness of the three improvement schemes on the YOLOv8n-obb algorithm, ablation experiments were conducted on the self-built dataset in a training environment. Model 1 represents the original YOLOv8n-obb object detection algorithm. Model 2 represents the use of C2f-Faster to replace the C2f layer in YOLOv8n-obb. Model 3 represents the introduction of the SimAM attention mechanism in the backbone network of YOLOv8n-obb. Model 4 represents replacing the default OBB detection head with the OBB-LSCD lightweight detection head. Model 5 introduces the SimAM attention mechanism into the backbone network based on Model 2. Model 6 uses the OBB-LSCD lightweight detection head based on Model 5.

Comparing models 1, 2, 3, and 4 in Table 1, it can be seen that after replacing C2f with C2f_Faster, model 2 reduces its parameter count by 0.71 M, GFLOPs by 2.6, while the P, R, and mAP50 metrics all show slight improvements. This indicates that C2f_Faster reduces the computational cost on redundant feature maps and has a stronger feature extraction capability than C2f. Compared to Model 1, Model 3's mAP50 has improved by 2.9% without an increase in the number of parameters. This indicates that the SimAM attention mechanism does not add extra parameters to the model, and after 3D weighting the feature maps, it enhances the model's attention to important features. Model 4 achieved an 8.7% improvement in Recall, which indicates that the introduction of shared convolution allows the model to learn features at different scales, and better adapt to objects of various sizes and different scenarios. Additionally, the number of parameters decreased by 0.71 M. In summary, using three improvement measures alone can improve its accuracy while lightweight the model, verifying the independent effectiveness of the three improvement measures.

Table 1: Ablation experiment table

Model	C2f-Faster	SimAM	OBB-LSCD	P (%)	R (%)	mAP50 (%)	Params (MB)	GFLOPs
1				90.5	78.3	89.7	3.08	8.3
2	√			91.3	80.4	90.9	2.37	6.6
3		√		92.6	84.2	92.6	3.08	8.3
4			√	92.4	87.0	92.2	2.38	6.6
5	√	√		91.8	88.1	93.9	2.37	6.6
6	√	√	√	92.1	89.9	94.8	1.67	4.8

Comparing models 1, 2, 5, and 6 in Table 1, it can be seen that after sequentially adding the C2f_Faster module, SimAM attention mechanism, and OBB-LSCD lightweight detection head to the baseline model, the mAP50 shows a stepwise increase. This indicates that the SimAM attention mechanism assigns higher

weights to important features in the C2f-Faster output feature maps, while the OBB-LSCD learns more general and higher-weighted important features, thereby enhancing the model's ability to recognize objects. In addition, the Recall of Model 6 has increased by 11.6% compared to Model 1, indicating that the model has improved the detection rate of objects in different drilling sites, which is of great significance for the working environment of frequent movement in underground drilling sites. At the same time, the gradual decrease in the number of parameters and GFLOPs demonstrates that the three improvements in this paper achieve the goal of model lightweighting while ensuring high object detection accuracy.

5.4 Attention Mechanism Comparison Experiment

To compare the impact of different attention mechanisms on model accuracy and parameter count, six sets of comparative experiments are designed, including SE, CBAM, CPCA, EMA, and MLCA. The network structure adopts the lightweight network structure proposed in this paper and only replaces the SimAM attention mechanism at the same position in the proposed algorithm with the above-mentioned attention mechanisms.

According to the analysis of the data in Table 2, it can be seen that in terms of precision, MLCA is 93.3%, which is the highest among the attention mechanisms in Table 2. SimAM's R and mAP50 are 89.9% and 94.8%, respectively, both higher than other attention mechanisms. In terms of lightweight, CBAM, CPCA, and EMA significantly increase the number of model parameters and reduce the inference speed of the model. The inference latency of CPCA is twice that of SimAM. Although the number of parameters of SE, MLCA, and SimAM is all on the order of 1.67 MB, SE, and MLCA introduce fully connected layers, pooling layers, and convolutional layers, which increase model inference latency by 2.7 and 4.5 ms, respectively, compared to SimAM. SimAM also increases the computation of energy functions and reduces the inference speed of the model, but it is still faster than other attention mechanisms in Table 2. In summary, the SimAM attention mechanism has significant advantages in both accuracy and speed compared to other attention mechanisms.

Table 2: Comparison experiments of different attention mechanism

Algorithm	P (%)	R (%)	mAP50 (%)	Params (MB)	GFLOPs	Speed (ms)
+SE	90.8	89.1	93.9	1.67	4.8	30.5
+CBAM	91.7	85.1	94.3	1.83	4.9	32.6
+CPCA	92.0	81.7	89.5	1.98	60	70.6
+EMA	92.9	86.8	93.3	1.70	5.2	37.5
+MLCA	93.3	89.3	94.5	1.67	4.8	32.3
+SimAM	92.1	89.9	94.8	1.67	4.8	27.8

5.5 Normalization Layer Comparison Experiments

To verify the impact of different normalization layers on model accuracy, four sets of experiments are designed in the training environment to compare the effects of different normalization layers on the model's P, R, and mAP50 metrics. The network structure is FSS-YOLO, only changing the normalization layer in the detection head.

The precision of the BN layer in Table 3 is the highest at 93.6%, which is 1.4%, 0.8%, and 0.4% higher than the GN, IN, and LN layers, respectively. The mAP50 of the IN layer is the same as that of the GN layer, both being 0.2% higher than the GN layer and 0.8% higher than the LN layer. The Recall of the IN layer is

0.3% higher than that of the GN layer. Based on the three metrics, it can be concluded that the IN layer has the greatest improvement. Therefore, all BN layers in the detection head are replaced with IN layers.

Table 3: Comparison results of different normalization layers

Normalization	P (%)	R	mAP50
BN	93.6	87.7	94.6
GN	92.1	89.6	94.8
LN	93.2	87.9	94.0
IN	92.8	89.9	94.8

5.6 YOLO Series Algorithm Comparison Experiments

For the rigor of the experiment, we selected three algorithms with built-in rotation object detection, YOLOv5n-obb, YOLOv8n-obb, and YOLOv11n-obb, for comparison on our self-built dataset.

Comparing the data in Table 4, it can be seen that in terms of accuracy, mAP50, is a key metric for evaluating the performance of the object detection algorithm. FSS-YOLO's mAP50 value reaches 94.8%, which is 5.1% higher than the original algorithm and higher than other compared algorithms. The improved YOLOv8n-obb is also the highest-performing algorithm in terms of Recall, with a Recall of 89.6%. But in terms of precision, YOLOv5n-obb is the highest-performing algorithm. In terms of lightweight, FSS-YOLO has 0.91 MB fewer parameters and 2.5 GFLOPs lower than YOLOv5n-obb. Compared to YOLOv11n-obb, it has 0.98 MB fewer parameters and 1.8 GFLOPs lower. In addition, the inference speed of the algorithm in this paper is 27.8 ms, making it the fastest inference speed algorithm in Table 4. To more intuitively demonstrate the differences in detection results among the algorithms, tests were conducted on three representative scenarios. As shown in Fig. 12, The first scenario is the image blur scenario, the second is the infrared photography scenario, and the third is the long-distance shooting scenario. According to the comparison of the detection results, FSS-YOLO can detect all targets, while YOLOv5n-obb, YOLOv8n-obb, and YOLOv11n-obb all showed missed detections.

Table 4: Comparison results of various algorithms

Algorithm	P (%)	R (%)	mAP50 (%)	Params (MB)	GFLOPs	Speed (ms)
MCIW-2 [29]	90.7	85.7	91.4	2.87	7.8	29.4
BPI-YOLO [30]	91.8	87.4	92.5	1.85	5.2	28.6
AMGC-YOLO [31]	93.6	84.6	91.3	1.54	3.3	26.7
TP-YOLO [32]	86.7	75.0	85.3	0.92	3.1	25.3
YOLOv5n-obb	92.3	78.4	90.0	2.58	7.3	28.2
YOLOv8n-obb	90.5	78.3	89.7	3.08	8.3	28.2
YOLOv11n-obb	91.1	82.5	91.9	2.65	6.6	28.6
FSS-YOLO	92.1	89.9	94.8	1.67	4.8	27.8

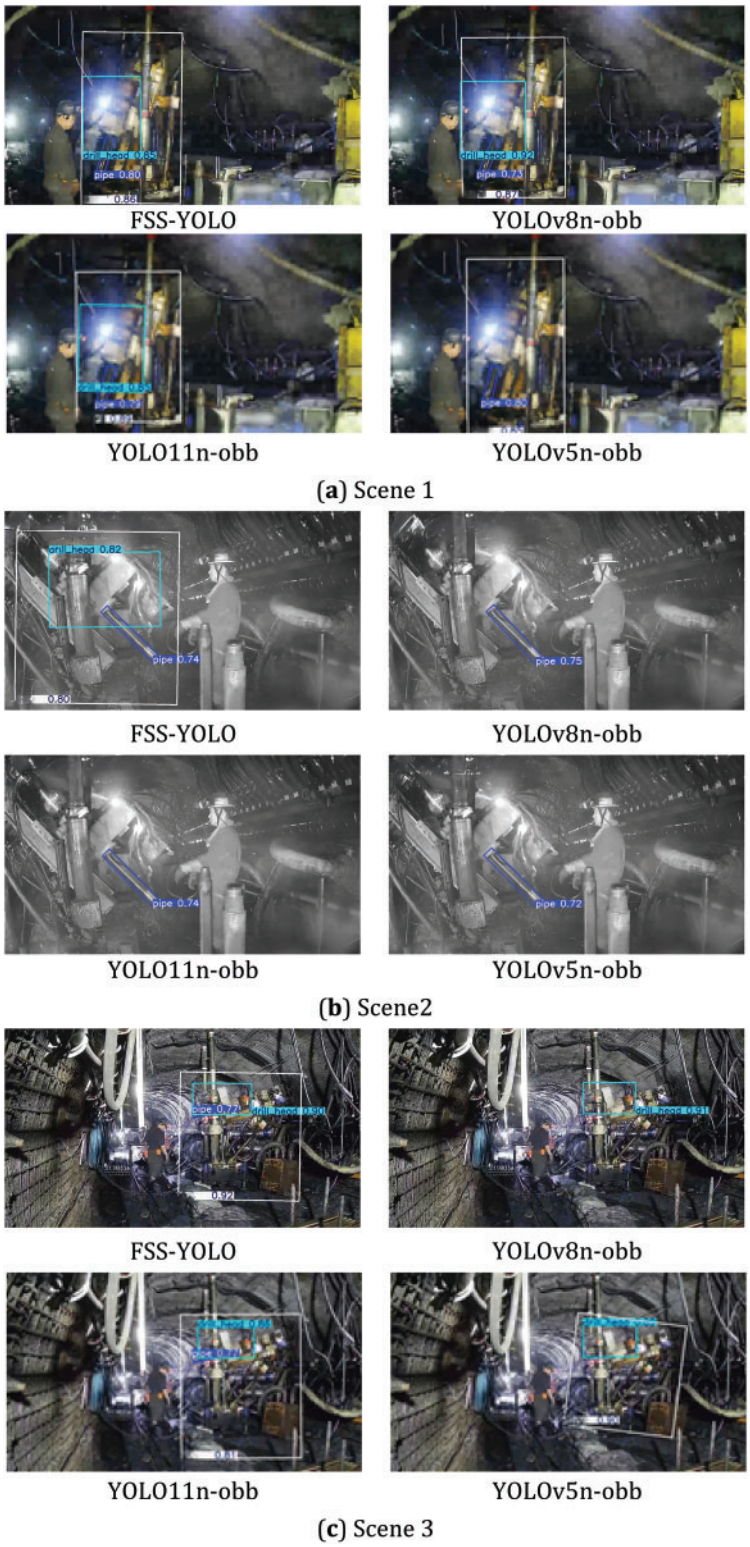


Figure 12: Comparison of visualization results. The representative scenes from top to bottom are blur, infrared photography, and long-distance shooting. White box represents drilling rig, The blue box represents the pipe. Blue-green box represents the drill_head

To highlight the contributions of our proposed algorithm, we also selected 4 different lightweight improved networks and trained them on our self-built dataset. The training results are shown in Table 4. MCIW-2 and BPI-YOLO have higher parameter counts and GFLOPs compared to the algorithms proposed in this paper, but they are slower in terms of speed. Although the precision of AMGC-YOLO is 1.5% higher than the algorithm proposed in this paper, its recall and mAP50 are 5.3% and 3.5% lower, respectively. The parameter counts of the proposed algorithm are close to twice that of TP-YOLO, and the speed is 2.5 ms slower, but the precision is much higher than TP-YOLO, especially the mAP50, which is 9.5% higher than TP-YOLO.

In summary, although the algorithm in this paper does not achieve an absolute advantage in terms of model parameter count and GFLOPs, it still outperforms most of the algorithms in Table 4. Furthermore, it demonstrates a decisive advantage in terms of accuracy, especially in the crucial mAP50 metric. Therefore, through comparative analysis, it can be concluded that the algorithm in this paper has a comprehensive advantage, achieving a good balance between model lightweight and accuracy.

The above experimental results demonstrate that FSS-YOLO, compared to other YOLO algorithms on the self-built dataset, has higher object recognition accuracy in different scenarios, fewer parameters and GFLOPs, and shorter inference time, showing comprehensive performance advantages. Moreover, the model can still achieve a speed of 27.8 ms per frame when deployed.

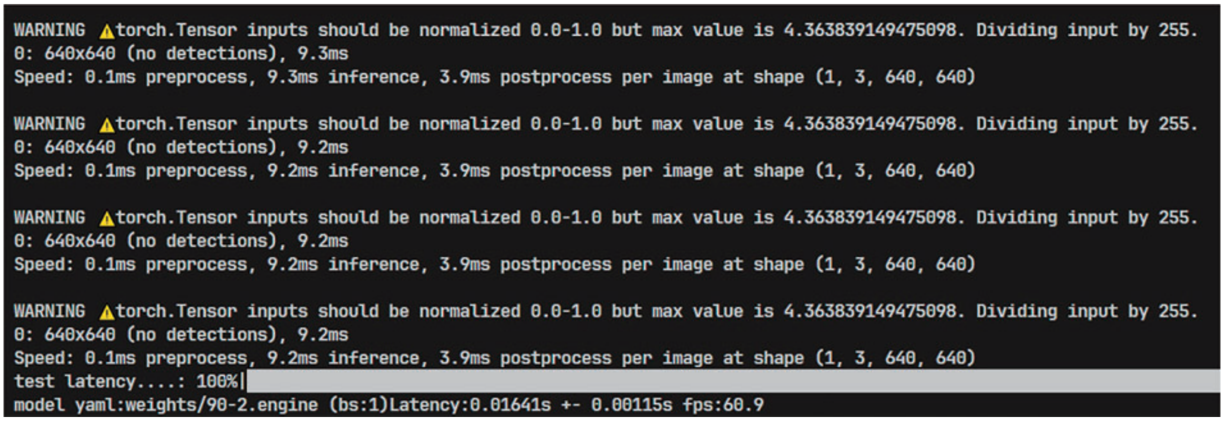
To eliminate accidental results in the experiment, this paper conducted 20 training sessions using 20 different random number seeds. The training results are shown in Table 5. The ranges of the P, R, and mAP50 of the baseline model are 0.894 to 0.912, 0.752 to 0.814, and 0.861 to 0.907, respectively, with the average values being 0.904, 0.787, and 0.891, respectively. The ranges of the P, R, and mAP50 of the model in this paper are 0.913 to 0.941, 0.873 to 0.906, and 0.911 to 0.952, respectively, with the average values being 0.927, 0.886, and 0.943, respectively.

Table 5: Statistical test results

Algorithm	P (%)	R (%)	mAP50 (%)	Mean (P/R/mAP50)
YOLOv8n-obb	89.4–91.2	75.2–81.4	86.1–90.7	90.4/78.7/89.1
FSS-YOLO	91.3–94.1	87.3–90.6	91.1–95.2	92.7/88.6/94.3

5.7 Deployment

To further accelerate the inference speed of the model, we quantized the weight file obtained from training from FP32 to INT8. As shown in Fig. 13, the speed of the model in this paper after INT8 quantization is approximately 9.3 ms, which is only one-third of that of the weight file with FP32. Moreover, to make it more convenient for application in coal mines, we have designed a platform for object detection in drilling sites. This platform uses the quantized weight file for inference and pushes the inference results to a web page through HTTP services. It can achieve parallel detection of multiple channels of video and display of the results. The details are shown in Fig. 14.



```
WARNING ⚠torch.Tensor inputs should be normalized 0.0-1.0 but max value is 4.363839149475098. Dividing input by 255.
0: 640x640 (no detections), 9.3ms
Speed: 0.1ms preprocess, 9.3ms inference, 3.9ms postprocess per image at shape (1, 3, 640, 640)

WARNING ⚠torch.Tensor inputs should be normalized 0.0-1.0 but max value is 4.363839149475098. Dividing input by 255.
0: 640x640 (no detections), 9.2ms
Speed: 0.1ms preprocess, 9.2ms inference, 3.9ms postprocess per image at shape (1, 3, 640, 640)

WARNING ⚠torch.Tensor inputs should be normalized 0.0-1.0 but max value is 4.363839149475098. Dividing input by 255.
0: 640x640 (no detections), 9.2ms
Speed: 0.1ms preprocess, 9.2ms inference, 3.9ms postprocess per image at shape (1, 3, 640, 640)

WARNING ⚠torch.Tensor inputs should be normalized 0.0-1.0 but max value is 4.363839149475098. Dividing input by 255.
0: 640x640 (no detections), 9.2ms
Speed: 0.1ms preprocess, 9.2ms inference, 3.9ms postprocess per image at shape (1, 3, 640, 640)
test latency....: 100%|
model yaml:weights/90-2.engine (bs:1)Latency:0.01641s +- 0.00115s fps:60.9
```

Figure 13: Inference time test

Figure 14: Web page diagram

6 Conclusions and Prospects

(1) This paper proposes FSS-YOLO, a lightweight drilling pipe detection algorithm based on YOLOv8n-obb, aimed at addressing the insufficient accuracy issues of the original YOLOv8n-obb algorithm. This paper' algorithm analyzes the problem of image redundancy during the creation of an underground dataset and uses C2f-Faster to reduce the computational cost of redundant feature maps and decrease the number of model parameters. The SimAM attention mechanism is introduced into the backbone network, enhancing the model's attention to important features without introducing additional parameters. Using shared convolution to optimize the default detection head, allowing the detection head to better learn feature maps at different scales, reducing the number of model parameters while improving its generalization ability. And replace all BN layers in the detection head with IN layers to further improve the recognition accuracy of the model.

(2) Compared to the original algorithm, FSS-YOLO achieves a 5.1% increase in mAP50, an 11.5% improvement in R, a 45.8% reduction in parameter count, and an inference speed of 27.8 ms per frame on the Nvidia Orin NX terminal device. Compared to other YOLO series algorithms, the proposed algorithm has advantages in both accuracy and speed. Meanwhile, the comparison experiment data and visualization results demonstrate that FSS-YOLO can quickly and accurately complete the drilling pipe detection task.

(3) Although FSS-YOLO uses a C2f-Faster module to alleviate the problem of redundant feature maps, the problem has not been truly solved. In future research, new methods need to be explored to further reduce the feature redundancy caused by high image similarity in the dataset.
(4) FSS-YOLO has only achieved fast and accurate detection, but it has not yet been translated into productivity. Therefore, in future research, FSS-YOLO should be used as a basis to replace other manual tasks, such as drill pipes counting and alerting for abnormal working conditions of drilling rigs.

Acknowledgement: The authors express their gratitude to Henan Polytechnic University, for administrative and technical support.

Funding Statement: The authors received no specific funding for this study.

Author Contributions: Mingyang Zhao is mainly responsible for manuscript writing. Miao Li is responsible for manuscript translation. Mingyang Zhao and Miao Li are jointly responsible for the analysis of experimental results. Xiaojun Li is responsible for the overall direction control of the paper. Bangbang Mu is responsible for video image processing and data annotation. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: Data will be made available on request.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

References

1. Cheng C, Hu Q, Luo Y, Wang B, Tao R, Sun Y. Experimental study of deformation evolution around the gas extraction borehole in a composite stratum. Fuel. 2025;381(5):133296. doi:10.1016/j.fuel.2024.133296.
2. Zhang S, Wang Z, Zhao Y, Chen D, Qiu Y, Wei J. Study on the design of gas extraction drilling angle of crossing coal layer drilling in the floor rock roadway considering the boreholes superposition effect. Fuel. 2024;378(19):132981. doi:10.1016/j.fuel.2024.132981.
3. Zhang K, Yang X, Xu L, Thé J, Tan Z, Yu H. Enhancing coal-gangue object detection using GAN-based data augmentation strategy with dual attention mechanism. Energy. 2024;287(2):129654. doi:10.1016/j.energy.2023.129654.
4. Wang GF, Ren HW, Zhao GR, Zhang DS, Wen ZG, Meng LY, et al. Research and practice of intelligent coal mine technology systems in China. Int J Coal Sci Technol. 2022;9(1):24. doi:10.1007/S40789-022-00491-3.
5. Yang YK, Zhou W, Jiskani IM, Wang ZM. Extracting unstructured roads for smart open-pit mines based on computer vision: implications for intelligent mining. Expert Syst Appl. 2024;249:123628. doi:10.1016/J.ESWA.2024.123628.
6. Yang T, Guo Y, Li D, Wang S. Vision-based obstacle detection in dangerous region of coal mine driverless rail electric locomotives. Measurement. 2025;239:115514. doi:10.1016/j.measurement.2024.115514.
7. Li W, Gao Z, Feng G, Hao R, Zhou Y, Chen Y, et al. Damage characteristics and YOLO automated crack detection of fissured rock masses under true-triaxial mining unloading conditions. Eng Fract Mech. 2025;314(1):110790. doi:10.1016/j.engfracmech.2024.110790.
8. Imam M, Baïna K, Tabii Y, Ressami EM, Adlaoui Y, Boufousse S, et al. Integrating real-time pose estimation and PPE detection with cutting-edge deep learning for enhanced safety and rescue operations in the mining industry. Neurocomputing. 2025;618(3):129080. doi:10.1016/j.neucom.2024.129080.
9. Ganesh N, Shankar R, Mahdal M, Murugan JS, Chohan J, Kalita K. Exploring deep learning methods for computer vision applications across multiple sectors: challenges and future trends. Comput Model Eng Sci. 2024;139(1):103–41. doi:10.32604/cmes.2023.028018.
10. Redmon J, Farhadi A. YOLO9000: better, faster, stronger. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2017 Jul 21–26; Honolulu, HI, USA. doi:10.48550/arXiv.1612.08242.

11. Redmon J. Farhadi AYO. LOv3: an incrementa improvement. arXiv:1804.02767. 2018.
12. Redmon J, Divvala S, Girshick R, Farhadi A. You only look once: unifed, real-time object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2016 Jun 27–30; Las Vegas, NV, USA. doi:10.1109/CVPR.2016.91.
13. Wang S, Zhu J, Li Z, Sun X, Wang G. GDPs-YOLO: an improved YOLOv8s for coal gangue detection. Int J Coal Prep Util. 2024;45(4):683–96. doi:10.1080/19392699.2024.2346626.
14. Liu Z, Miao S, Liang Y, Li J, Meng P, Sun T. Research on coal gangue recognition method based on SFD-YOLOv5s. Int J Coal Prep Util. 2024;1–16. doi:10.1080/19392699.2024.2428640.
15. Shan P, Meng Z, Xu H, Li C, Zhang L, Xi B. Research on accurate recognition and refuse rate calculation of coal and gangue based on thermal imaging of transporting situation. Measurement. 2025;244(1):116574. doi:10.1016/j.measurement.2024.116574.
16. Zhang M, Cao Y, Jiang K, Li M, Liu L, Yu Y, et al. Proactive measures to prevent conveyor belt failures: deep learning-based faster foreign object detection. Eng Fail Anal. 2022;141(10):106653. doi:10.1016/j.engfailanal.2022.106653.
17. Ling J, Fu Z, Yuan X. Lightweight coal mine conveyor belt foreign object detection based on improved Yolov8n. Sci Rep. 2025;15(1):10361. doi:10.1038/S41598-025-87848-1.
18. Ren H, Fan A, Zhao J, Song H, Wen Z, Lu S. A dynamic weighted feature fusion lightweight algorithm for safety helmet detection based on YOLOv8. Measurement. 2025;253(3):117572. doi:10.1016/J.MEASUREMENT.2025.117572.
19. Zhang L, Ma H, Huang J, Zhang C, Gao X. An improved lightweight safety helmet detection algorithm for YOLOv8. Comput Mater Contin. 2025;83(2):2245–65. doi:10.32604/CMC.2025.061519.
20. Li J, Xie S, Zhou X, Zhang L, Li X. Real-time detection of coal mine safety helmet based on improved YOLOv8. J Real-Time Image Process. 2024;22(1):26. doi:10.1007/s11554-024-01604-8.
21. Dong X, Wang X, Li B, Wang H, Chen G, Cai M. YH-Pose: human pose estimation in complex coal mine scenarios. Eng Appl Artif Intell. 2024;127(6):107338. doi:10.1016/j.engappai.2023.107338.
22. Jin H, Ren S, Li S, Liu W. Research on mine personnel target detection method based on improved YOLOv8. Measurement. 2025;245(2):116624. doi:10.1016/j.measurement.2024.116624.
23. Shao X, Liu S, Li X, Lyu Z, Li H. Rep-YOLO: an efficient detection method for mine personnel. J Real-Time Image Process. 2025;21(2):28. doi:10.1007/s11554-023-01407-3.
24. Shao X, Li X, Yang T, Yang Y, Liu S, Yuan Z. Underground personnel detection and tracking based on improved YOLOv5s and DeepSORT. Coal Sci Technol. 2023;51(10):291–301. (In Chinese). doi:10.13199/j.cnki.cst.2022-1933.
25. Xu S, Jiang W, Liu Q, Wang H, Zhang J, Li J, et al. Coal-rock interface real-time recognition based on the improved YOLO detection and bilateral segmentation network. Undergr Space. 2025;21:22–43. doi:10.1016/j.undsp.2024.07.003.
26. Zhang D, Jiang Y. Drill pipe counting method based on improved MobileNetV2. J Mine Autom. 2022;48(10):69–75. (In Chinese). doi:10.13272/j.issn.1671-251x.2022060019.
27. Gao R, Hao L, Liu B, Wen J, Chen Y. Research on underground drill pipe counting method based on improved ResNet network. J Mine Autom. 2020;46(10):32–7. (In Chinese). doi:10.13272/j.issn.1671-251x.2020040054.
28. Yao C, Hu Y. Drilling pipe counting algorithm based on video analysis in coal mine. Coal Technol. 2023;42(8):203–6. (In Chinese). doi:10.13301/j.cnki.ct.2023.08.044.
29. Jiang Y, Liu S. A coal mine underground drill pipes counting method based on improved YOLOv8n. J Mine Autom. 2024;50(8):112–9. (In Chinese). doi:10.13272/j.issn.1671-251x.2024040073.
30. Wu L, Chen L, Lyu Y. A lightweight-based method for real-time monitoring of lump coal on conveyor belts. Coal Sci Technol. 2023;51(S2):285–93. (In Chinese). doi:10.12438/cst.2023-121710.21203/rs.3.rs-3031808/v1.
31. Ruan S, Wang J, Gu Q. Research on mining area obstacle detection model for edge computing. Coal Sci Technol. 2024;52(11):141–52. (In Chinese). doi:10.12438/cst.2024-0664.
32. Yue Z, Jin Q, Pan S. Intelligent detection method of lightweight blasthole based on deep learning. J China Coal Soc. 2024;49(5):2247–56. (In Chinese). doi:10.13225/j.cnki.jccs.2023.0557.

33. Du YR, Han YP, Su YH, Wang JX. A lightweight model based on you only look once for pomegranate before fruit thinning in complex environment. Eng Appl Artif Intell. 2024;137:109123. doi:10.1016/j.engappai.2024.109123.
34. Chen J, Kao S, He H, Zhuo W, Wen S, Lee C-H, et al. Run, don't walk: chasing higher FLOPS for faster neural networks. arXiv:2303.03667. 2023. doi:10.1109/CVPR52729.2023.01157.
35. Shi H, Yang W, Chen D, Wang M. CPA-YOLOv7: contextual and pyramid attention-based improvement of YOLOv7 for drones scene target detection. J Vis Commun Image Represent. 2023;97(3):103965. doi:10.1016/j.jvcir.2023.103965.
36. Wang D, Tan J, Wang H, Kong L, Zhang C, Pan D, et al. SDS-YOLO: an improved vibratory position detection algorithm based on YOLOv11. Measurement. 2025;244(6):116518. doi:10.1016/j.measurement.2024.116518.
37. Hu J, Shen L, Sun G. Squeeze-and-excitation networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2018 Jun 18–23; Salt Lake City, UT, USA. doi:10.48550/arXiv.1709.01507.
38. Woo S, Park J, Lee J-Y, Kweon IS. CBAM: convolutional block attention module. In: Proceedings of the European Conference on Computer Vision (ECCV); 2018 Sep 8–14; Munich, Germany. doi:10.48550/arXiv.1807.06521.
39. Huang H, Chen Z, Zou Y, Lu M, Chen C. Channel prior convolutional attention for medical image segmentation. Comput Biol Med. 2024;178(27):108784. doi:10.1016/j.compbiomed.2024.108784.
40. Yang L, Zhang RY, Li L, Xie X. SimAM: a simple, parameter-free attention module for convolutional neural networks. In: Proceedings of the 38th International Conference on Machine Learning. 2021; PMLR. Vol. 139. p. 11863–74.