Advances in Computer, Signals and Systems (2023)
Clausius Scientific Press, Canada

DOI: 10.23977/acss.2023.071015
ISSN 2371-8838 Vol. 7 Num. 10

Improved Lightweight Rebar Detection Network Based on YOLOv8s Algorithm

Hu Zhichao, Wang Yi, Wu Junping, Xiong Wanli, Li Bilian

Computer Science and Engineering College, Sichuan University of Science & Engineering, Zigong, Sichuan, 643000, China

Keywords: Yolov8; SPDConv; Bi-PAN-FPN; Inner-IoU; Dynamic Head

Abstract: This paper introduces an improved YOLOv8-based algorithm, Yolo-Rebar, designed to address the challenges of intelligent rebar counting in construction engineering. By integrating SPDConv to replace traditional convolution for downsampling, and combining bi-directional feature pyramid networks (Bi-PAN-FPN), internal intersection over union (Inner-IoU) evaluation strategy, and Dynamic Head component, Yolo-Rebar optimizes the network structure and inference process, significantly reducing computational load and parameter count while maintaining a high detection accuracy (mAP of 0.985). Maintaining a low computational demand (29.9 GFLOPs) and a moderate model size (27.1 MB), Yolo-Rebar outperforms Yolov5s and Yolov8s models in detection accuracy by 1.5% and 0.8% respectively, and compared to Yolov3-app and Yolov8m models, it requires lower computational resources while maintaining high accuracy. Empirical results demonstrate that Yolo-Rebar exhibits remarkable robustness and precision in complex construction environments, such as varying lighting conditions, rebar stacking, and occlusions. This research not only enhances the efficiency and accuracy of material acceptance in construction engineering but also provides a new direction for the further development of deep learning technology in industrial applications.

1. Introduction

In construction engineering of concrete structures, the quantity of threaded steel entering the construction site is enormous. Material acceptance at the site, as the first link in construction, directly affects the speed and quality of infrastructure development. However, most construction sites still rely on manual counting for threaded steel quantity acceptance. Manual counting is inefficient, labor-intensive for workers, prone to visual fatigue, and often leads to inaccuracies. In recent years, with the rapid development of deep learning technology, object detection has made significant progress in the field of computer vision and has been widely applied in various fields, including but not limited to autonomous driving[1], security surveillance[2], and smart homes[3].

Currently, deep learning has become the mainstream method in the field of object detection, achieving significant results on large-scale datasets. Zhao[4] et al. used median filtering for denoising, improved OTSU algorithm and Sabel operator, and center aggregation algorithm for edge information extraction and segmentation, and frame-by-frame counting of images, achieving an accuracy of 96%. GHAZALI[5] et al. proposed a method to count using image contours, employing Gaussian blur and

morphological closing for image preprocessing, combined with Hough transform, LoG, and morphological operations to create contours of rebar, and then calculated the number of different contours to obtain the count of steel tubes. This method accurately counts circular and rectangular steel tubes. Shin Y[6] et al. used a network combining CNN and homography for estimating dimensions and counting rebar end images. Fan Z[7] et al. proposed a CNN-DC convolutional network framework, using CNN for candidate center point detection, followed by distance clustering to cluster candidate center points and locate the center of rebars, ultimately achieving 99.76% accuracy. Li[8] et al. improved the precision of rebar end face recognition in the YOLOv3 framework by adding FPN layers, introducing IoU loss, and Focal loss. Zheng[9] et al. improved the YOLOv5 model by introducing the CA attention mechanism and SD_IoU Loss function, effectively solving the problem of dense and sticky bundled rebar recognition, achieving an average accuracy of 97.9% on their custom dataset. Li[10] et al., in drone recognition tasks, modified the backbone and neck network of YOLOv8, borrowing the idea of Bi-FAN-FPN, adding downsampling in FPN, and replacing some C2f modules with GhostblockV2 to reduce information loss due to long-distance transmission and significantly decrease the model parameter count. They used WiseIoU loss as the bounding box regression loss. Their model surpassed the original in both accuracy and speed on the public dataset VisDrone2019.

This paper proposes a rebar end face image detection network model (Yolo-Rebar) based on the Yolov8 model, combined with SPDConv, Bi FPN, Inner-IoU, Dynamic Head, and other technologies. This model significantly improves average precision while maintaining a similar volume, demonstrating great potential in practical engineering applications.

2. Model Improvement and Optimization

2.1 Improved Backbone Network

2.1.1 Adopting SPDConv in Place of Conv

In YOLOv8, the Conv module effectively extracts and transforms the information of input features through the combination of convolution, batch normalization, and SiLU activation function. The structure of the Conv module is illustrated as shown in the figure 1 below:

Figure 1: Yolov8 Standard Conv Module Structure Diagram

However, when processing small object detection, the standard Conv Module may lead to information loss due to reducing spatial dimensions. This can negatively impact recognition accuracy. To address this issue, we introduce SPDConv to replace Conv for feature extraction.

SPDConv (Space-to-depth Conv) is an innovative Convolutional Neural Network (CNN) module designed specifically for handling low-resolution images and small objects (figure 2)[11].

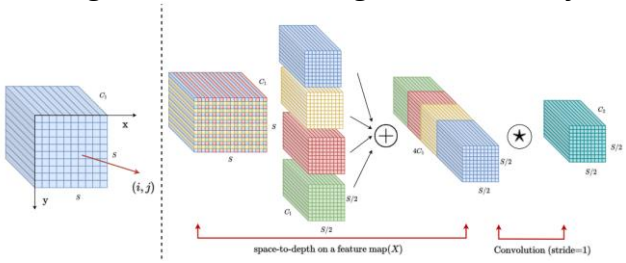


Figure 2: The implementation schematic of SPDConv[11]

2.1.2 Adopting SPPF-LSKA in Place of SPPF

Spatial Pyramid Pooling (SPP) [12]is a technique used in deep learning to address varying sizes of input images. The SPP structure is illustrated in Figure 3: implemented by adding a pooling layer after the convolutional layer, this layer divides the feature map into different regions and performs pooling (typically max pooling) on each region, generating fixed-length outputs to ensure consistency regardless of the input feature map's size. In YOLOv5, the incorporation of the SPP module greatly enhances the network's feature extraction capabilities. It not only enables YOLOv5 to handle images of different sizes but also improves the model's adaptability to size variations.

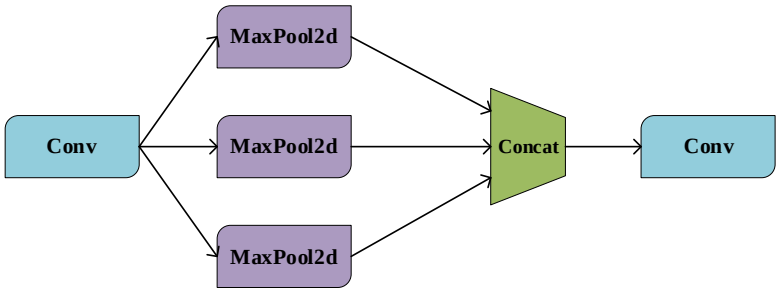


Figure 3: Yolov5 SPP Module Structure Diagram

YOLOv8 introduces the SPPF module (Spatial Pyramid Pooling - Fast), an optimization of the traditional Spatial Pyramid Pooling (SPP). The structural diagram is depicted in Figure 4: SPPF significantly reduces computational complexity by employing efficient pooling strategies, including a reduction in the number of pooling layers and the consolidation of multiple pooling operations.

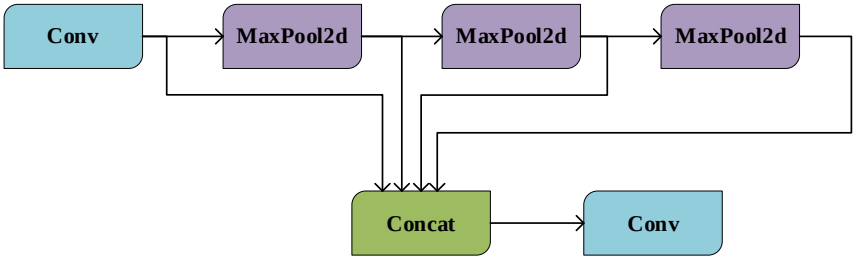


Figure 4: Yolov5 SPPF Module Structure Diagram

To further enhance spatial awareness and improve computational efficiency, this paper introduces Large Separable Kernel Attention (LSKA) into the SPPF structure. LSKA[13] is an attention mechanism based on a large-scale separable kernel, further optimizing the Large Kernel Attention[14] (LKA). The detailed illustration of LSKA is shown in Figure 5.

2.2 Improved Neck Network

YOLOv8 utilizes the PAN-FPN structure, based on PANet[15], to enhance the feature pyramid of FPN[16] and improve semantic capabilities. However, PAN-FPN still needs improvements in small object detection. To address this, Bi-PAN-FPN[17] has been introduced. It improves network structure and adds inter-layer connections, enhancing feature fusion efficiency. This compensates for the limitations of FPN and PANet, thereby improving detection performance for small objects.

In this paper, building upon Bi-PAN-FPN, the fusion of the P2 and P3 layers with the last layer of FPN is performed to enhance the utilization of large-sized feature information. Additionally, the original concatenation module in YOLOv8 is replaced with the BiFPN (Bi-directional Feature Pyramid Network) module. Compared to the Concat module, the BiFPN module achieves a more

complex feature fusion through operations such as weight learning and normalization. Considering the increase in computational load, the output channel number in the neck network is uniformly set to 256 in this study. Experimental results demonstrate that this network model can maintain average precision in recognition even with a significant reduction in computational parameters. The computation process of the BiFPN module is as follows:

Assuming there are N input feature tensors forming the input list $X = [x_1, x_2, ..., x_N]$, along with an equally long fusion weight vector $W = [w_1, w_2, ..., w_N]$ where w_i is a learnable parameter, the BiFPN module performs the following computation:

Firstly, for each weight w_i, the ReLU function is applied to ensure its non-negativity:

$$\mathrm{Re}\,LU(w_i) = \max(0, w_i) \tag{1}$$

Then, these non-negative weights are normalized, undergoing weight normalization:

$$W' = \frac{W}{\sum_{i=1}^{N} w_i} \tag{2}$$

Finally, the weighted sum is computed by multiplying each input feature tensor with its corresponding weight and then summing them all:

$$Y = \sum_{i=1}^{N} w_i' \cdot x_i \tag{3}$$

2.3 Improved Head Network

This paper introduces DyHead[18](Dynamic Head) . as the detection head of the network. DyHead provides a unified object detection framework that significantly enhances the representation capability of the object detection head by combining various self-attention mechanisms, without incurring additional computational costs. Details of DyHead are illustrated in Figure 5.

Typically, the formula for self-attention mechanisms is often represented as shown in Equation (4):

$$W(F) = \pi(F) \cdot F, F \in R^{L \times S \times C} \tag{4}$$

where $\pi(\cdot)$ is an attention function, and L, S, C represent the Level, Scale, and Channel dimensions, respectively. In DyHead, to overcome the high computational cost associated with directly operating on these three dimensions, a decomposition strategy is employed. This strategy decomposes operations on the L, S, and C dimensions, and through experimentation, it was found that the decomposed operations precisely correspond to the attention in three aspects: Scale-aware, Spatial-aware, and Task-aware. Subsequently, these operations are recombined in a cascading manner. This approach significantly improves attention in each dimension while substantially reducing computational complexity. This method effectively optimizes the efficiency and performance of the self-attention mechanism.

$$W(F) = \pi_C(\pi_S(\pi_L(F) \cdot F) \cdot F) \cdot F \tag{5}$$

Here, $\pi_C(\cdot)$, $\pi_S(\cdot)$, $\pi_L(\cdot)$ respectively, represent attention functions applied to the L, S, and C dimensions: Scale-aware Attention, Spatial-aware Attention, Task-aware Attention.

2.4 Improved Loss Function

Yolov8's bounding box (bbox) regression uses the Complete Intersection over Union (CIoU) as the loss function. The calculation formula for CIoU loss is as follows:

$$L_{CIoU} = IoU - \frac{\rho^2(b, b^{gt})}{c^2} - \alpha \cdot \upsilon \tag{6}$$

Where IoU is the Intersection over Union, $\rho^2(b, b^{gt})$ is the Euclidean distance between the predicted bounding box b and the real bounding box(b^{gt})'s center point, c is the diagonal length of the minimum enclosing region containing both bounding boxes, and α is a weight parameter used to balance the influence of aspect ratio.

CIoU is an improvement over traditional Intersection over Union (IoU). It not only considers the overlapping region of bounding boxes but also takes into account the distance between the center points of the bounding boxes and the similarity of their aspect ratios. This gives it a more comprehensive measure of similarity. However, CIoU does not consider the rationality of IoU loss itself, which to some extent determines the quality of the detection results. To address this issue, this paper uses Inner-CIoU to replace CIoU.

Inner-CIoU loss can be obtained through IoU, L_{CIoU} and IoU^{inner} where:

$$L_{inner}\text{-}CIoU = L_{CIoU} + IoU - IoU^{inner} \tag{7}$$

Inner-IoU[19]. Introduces auxiliary bounding boxes to calculate IoU loss. It introduces a scale factor (ratio) to control the size of the auxiliary bounding boxes. By distinguishing different regression samples and using auxiliary bounding boxes of different scales to calculate the loss, it accelerates the regression process of bounding boxes. The calculation formula is as follows:

$$b_l^{gt} = x_c^{gt} - \frac{w^{gt} * ratio}{2}, b_r^{gt} = x_c^{gt} + \frac{w^{gt} * ratio}{2}, \quad b_t^{gt} = y_c^{gt} - \frac{h^{gt} * ratio}{2}, b_b^{gt} = y_c^{gt} + \frac{h^{gt} * ratio}{2} \tag{8}$$

$$b_l = x_c - \frac{w * ratio}{2}, b_r = x_c + \frac{w * ratio}{2}, \quad b_t = y_c - \frac{h * ratio}{2}, b_b = y_c + \frac{h * ratio}{2} \tag{9}$$

Where (x_c^{gt}, y_c^{gt}) represents the center point of the ground truth box (GT Box) and the center point of the inner ground truth box, (x_c, y_c) represents the center point of the anchor point and the center point of the inner anchor point. The width and height of the ground truth box are w^{gt} and h^{gt}, and the width and height of the anchor point are w and h. $b_l^{gt}, b_r^{gt}, b_t^{gt}, b_b^{gt}$ respectively, represent the left, right, upper, and lower boundaries of the ground truth box, calculated from the center point of the ground truth box and the scaling factor ratio. b_l, b_r, b_t, b_b respectively, represent the left, right, upper, and lower boundaries of the anchor point, calculated from the center point of the anchor point and the scaling factor ratio.

$$\text{int}\,er = (\min(b_r^{gt}, b_r) - \max(b_l^{gt}, b_l)) * (\min(b_b^{gt}, b_b) - \max(b_t^{gt} - \max(b_t^{gt}, b_t)) \tag{10}$$

$$union = (w^{gt} * h^{gt}) * (ratio)^2 + (w * h) * (ratio)^2 - \text{int}\,er \tag{11}$$

$$IoU^{inner} = \frac{inter}{union} \tag{12}$$

Where inter represents the intersection area of the GT box and the anchor box, and union is the union area of the GT box and the anchor box.

2.5 Improved Network Structure

Thus, the improved reinforcement bar end-face image detection model based on YOLOv8 is shown in Figure 5. Compared to the original YOLOv8, improvements have been made in the backbone, neck, head, and loss functions, with specific changes indicated by the graphical labels in Figure 5.

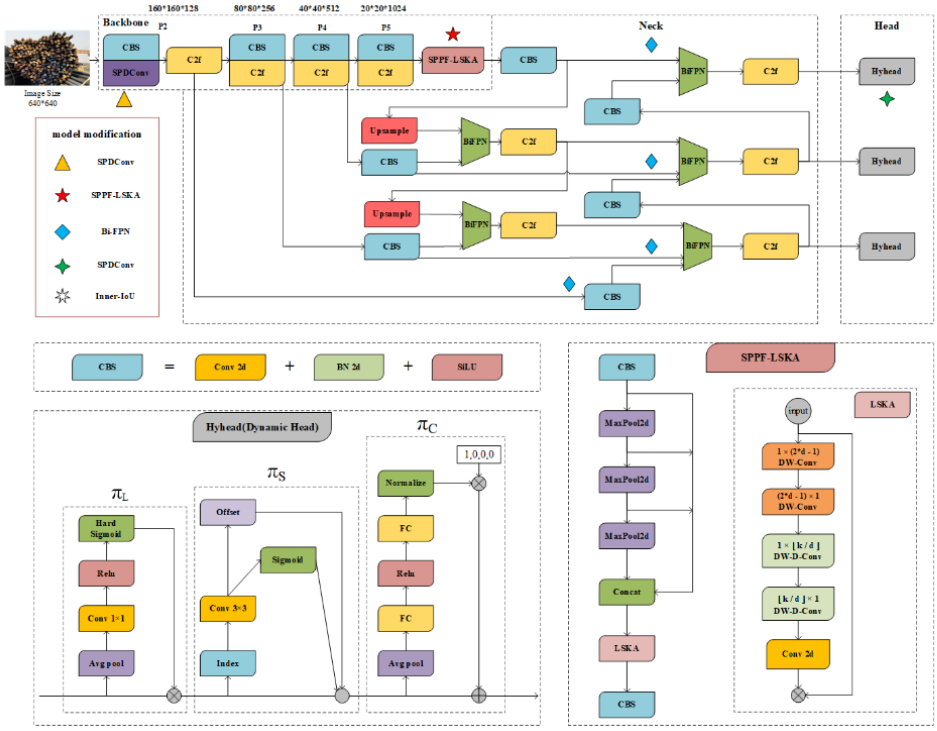


Figure 5: Improved network architecture for reinforcement bar end-face image detection based on YOLOv8

3. Experiment

3.1 Experimental Environment

The computer operating system used for this experiment is Ubuntu 18.04.5 LTS, with a 64-bit system. The processor model is Intel® Xeon(R) Platinum 8176 CPU @ 2.10GHz × 112. The device has a memory capacity of 187.5GB. There are two NVIDIA 2080ti GPUs, each with 11GB of VRAM, totaling 22GB of GPU memory. The GPU driver version is 510.108.03, and the CUDA version is 11.6. The YOLOv8 model is implemented using the PyTorch deep learning framework, with torch version 2.0.

3.2 Dataset and Its Preprocessing

This study utilized the 'Smart Inventory - Reinforcement Steel Artificial Intelligence Identification Competition' provided thread steel dataset and a portion of self-annotated datasets for experimental validation, totaling 450 images. The selected dataset covers various scenarios of reinforcement bar end-face images, including different lighting conditions, irregular arrangements of steel bars, voids, mutual occlusions, and stacking, to ensure that the experiments cover various challenges in real-world reinforcement detection. Some examples of the data are shown in the Figure 6.

Figure 6: Examples of images from the reinforcement bar dataset

The dataset contains only one category, which is the steel reinforcement. The entire dataset was divided into a training set (200 images), a validation set (50 images), and a test set (200 images). The important parameters for the training process are shown in Table . Considering the limited size of the training dataset, to ensure the model's generalization and robustness, this paper employed data augmentation techniques such as Mosaic, flipping, and translation during the training process to expand the dataset. The specific configuration parameters for data augmentation are shown in Table.1.

Table 1: Table of Training Parameter Settings

Parameter	Value
Epochs	100
Batch Size	16
Optimizer	SGD
NMS IoU	0.7
Initial Learning Rate	1 ×10-2
Final Learning Rate	1 ×10-2
Momentum	0.973
Weight Decay	5 ×10-4
Image Scale	0.5
Image Flip Left-Right	0.5
Mosaic	1.0
Ratio(Inner IoU)	1.2
Close Mosaic	Last 10 epochs

3.3 Model Evaluation Indices

This paper will use mAP (mean Average Precision) as a reference metric for the network's performance, which is a widely adopted evaluation method in the field of object detection. mAP evaluates the overall performance of the model by considering both precision and recall. Specifically, it calculates the average precision (AP) for each class, which is the area under the precision-recall curve (P-R curve), and then takes the average of these AP values to obtain mAP. The calculation

formula is as follows:

$$mAP = \frac{1}{n}\sum_{i=1}^{n} AP_i \tag{13}$$

Where n is the number of categories in the dataset. In this paper, the dataset contains only one category, which is steel reinforcement, so mAP=AP. The formulas for precision and recall are as follows:

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

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

In equations (14) and (15), TP represents true positives, FP represents false positives, and FN represents false negatives.

3.4 Ablation Experiment

The reinforcement bar end-face image detection model designed in this paper primarily improves the neck, backbone, and head of the baseline model (Yolov8-s), and replaces CIoU with Inner-CIoU in the final loss function as the bounding box regression loss. Through conducting ablation experiments to systematically analyze the improvement of each module on model performance, the average precision mean mAP is used as the evaluation metric in this experiment.

Table 2: Results of ablation experiments

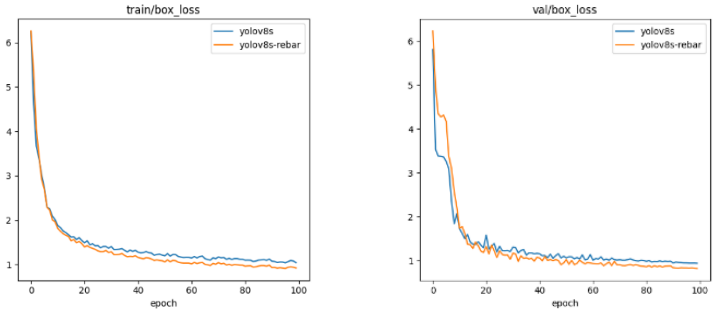
Model	Comput-ational /GFLOPs	Weight/ MB	Introduci-ng SPDCon-v	Introduci-ng SPPF-LSKA	Modifyi-ng the feature pyramid	Introduc-eng DyHead detection head	Mod-fying the loss funct-ion	mAP
Baseline	28.8	22.5	×	×	×	×	×	0.977
Model 1	31.6	27.0	√	×	×	×	×	0.978
Model 2	29.7	24.7	×	√	×	×	×	0.982
Model 3	25.4	15.0	×	×	√	×	×	0.977
Model 4	28.5	22.0	×	×	×	√	×	0.981
Model 5	28.8	22.5	×	×	×	×	√	0.979
Yolo-Rebar	29.9	27.1	√	√	√	√	√	0.985

Through comparative analysis using ablation experiments, we found that each module implemented in this paper positively impacts the baseline model. The optimized Yolo-Rebar model size increased by 4.6MB compared to the original, but its accuracy improved by 0.8%, with only a 1.1 GFLOPs increase in computational complexity.

3.5 Experimental Curve Comparison

This paper trained the baseline model (Yolov8s) and the proposed optimized model (Yolo-Rebar) on the rebar dataset according to the parameters set in Table . During the statistical detection process, the model's weights and mAP (mean average precision) metrics were recorded. The comparison of loss curves for both models on the training and validation sets is shown in figure 7 (a), (b). From the

curves, it can be observed that the Inner-ICoU used in this paper has lower loss values on both the training and validation sets, and the loss curves are smoother.



Training set loss curve validation set loss curve

Figure 7: Model training curve loss function compariso

3.6 Deep Learning Model Performance Comparison Experiment

Based on the different anchor generation mechanisms, deep learning methods in the field of object detection are mainly divided into one-stage methods and two-stage methods. This paper selected the well-known Faster R-CNN in the two-stage object detection domain and the leading YOLO series frameworks in the one-stage object detection domain, including YOLOv3-SPP, YOLOv5s, YOLOv8s. To highlight the superiority of the proposed model, YOLOv8m was also chosen for comparison in the experiments. From, it can be seen that YOLOv8-Rebar achieves the highest detection accuracy (mAP of 0.985) while maintaining low computational complexity (29.9 GFLOPs) and moderate model size (27.1 MB). Compared to other models such as YOLOv3-APP and YOLOv8m, which also demonstrate high detection accuracy, YOLOv8-Rebar achieves a balance between accuracy, computational complexity, and model size. In comparison to YOLOv5s and YOLOv8s, although YOLOv8-Rebar has a slightly higher model size, it outperforms them in average precision by 1.5% and 0.8%, respectively. This demonstrates that YOLOv8-Rebar exhibits excellent overall performance in terms of accuracy, computational complexity, and model size" (See table 3).

Table 3: The experimental results of multiple models are compared

Model	Computational /GFLOPs	Weight/MB	mAP
Faster-RCNN	-	318	0.727
Yolov3-app	284.2	209.9	0.984
Yolov5s	24.2	18.5	0.970
Yolov8s	28.8	22.5	0.977
Yolov8m	79.1	52.0	0.983
Yolov8-Rebar	29.9	27.1	0.985

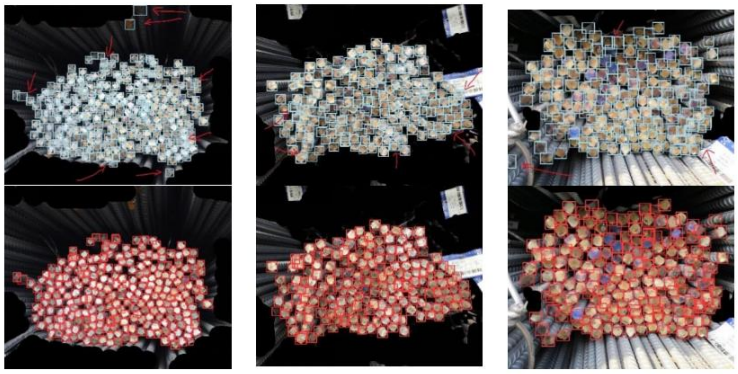
3.7 Analysis of Model Test Results

To visually demonstrate the performance of Yolo-Rebar, the trained model is tested on the test set and compared with the baseline model Yolov8s. The results are shown in Figure . In Figure , the light blue boxes represent the detection results of the baseline model Yolov8s, and the red boxes represent the detection results of Yolo-Rebar. The red arrows in Yolov8s indicate cases of misidentification and missed detection during the detection process, while in Yolo-Rebar, the recognition results in the same areas are good, with almost no cases of missed or misidentified objects. Additionally, in

situations with densely distributed and stacked rebars, Yolo-Rebar can accurately detect the positions of rebars.

4. Conclusion

Yolo-Rebar algorithm in this paper effectively improves the existing YOLOv8 algorithm, especially for the detection of rebar end faces in complex scenarios. Integrated with technologies such as SPDConv, Bi-PAN-FPN, Inner-IoU, and Dynamic Head, Yolo-Rebar demonstrates outstanding performance in handling high-density and multi-layered stacked rebars. This algorithm significantly reduces computational resource consumption while maintaining high processing speed, making it suitable for real-time detection applications. Yolo-Rebar holds significant value for the automation and accuracy improvement of rebar detection in construction engineering, showcasing the potential of deep learning in industrial applications.



(a) (b) (c)

Figure 8: Results of model testing

References

[1] Alaba S, Ball J. Deep Learning-Based Image 3-D Object Detection for Autonomous Driving: Review, IEEE Sensors Journal. 2023, 23(4): 3378-3394.

[2] Bitirgen K, Filik Ü B. A hybrid deep learning model for discrimination of physical disturbance and cyber-attack detection in smart grid, International Journal of Critical Infrastructure Protection. 2023, 40: 100582.

[3] Stephan M, Santra A. Radar-based human target detection using deep residual u-net for smart home applications, 2019 18th IEEE international conference on machine learning and applications (ICMLA), IEEE. 2019, 175-182.

[4] Zhao J, Xia X, Wang H, et al. Design of real-time steel bars recognition system based on machine vision, 2016 8th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC), IEEE. 2016, 1: 505-509.

[5] Ghazali M F, Wong L K, See J. Automatic detection and counting of circular and rectangular steel bars, 9th International Conference on Robotic, Vision, Signal Processing and Power Applications: Empowering Research and Innovation, Springer Singapore. 2017: 199-207.

[6] Shin Y, Heo S, Han S, et al. An image-based steel rebar size estimation and counting method using a convolutional neural network combined with homography, Buildings. 2021, 11(10): 463.

[7] Fan Z, Lu J, Qiu B, et al. Automated steel bar counting and center localization with convolutional neural networks, arXiv preprint arXiv. 2019,1906.00891.

[8] Li Y, Lu Y, Chen J. A deep learning approach for real-time rebar counting on the construction site based on YOLOv3 detector, Automation in Construction. 2021, 124: 103602.

[9] Zheng Y, Zhou G, Lu B. A Multi-Scale Rebar Detection Network with an Embedded Attention Mechanism, Applied Sciences. 2023, 13(14): 8233.

[10] Li Y, Fan Q, Huang H, et al. A Modified YOLOv8 Detection Network for UAV Aerial Image Recognition, Drones. 2023, 7(5): 304.

[11] Sunkara R, Luo T. No more strided convolutions or pooling: A new CNN building block for low-resolution images and small objects, Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Cham:

Springer Nature Switzerland. 2022, 443-459.
[12] He K, Zhang X, Ren S, et al. Spatial pyramid pooling in deep convolutional networks for visual recognition, IEEE transactions on pattern analysis and machine intelligence. 2015, 37(9): 1904-1916.
[13] Lau K W, Po L M, Rehman Y A U. Large Separable Kernel Attention: Rethinking the Large Kernel Attention Design in CNN, Expert Systems with Applications. 2024, 236: 121352.
[14] Guo M H, Lu C Z, Liu Z N, et al. Visual attention network, Computational Visual Media. 2023, 9(4): 733-752.
[15] Liu S, Qi L, Qin H, et al. Path aggregation network for instance segmentation, Proceedings of the IEEE conference on computer vision and pattern recognition. 2018, 8759-8768.
[16] Lin T Y, Dollár P, Girshick R, et al. Feature pyramid networks for object detection, Proceedings of the IEEE conference on computer vision and pattern recognition. 2017, 2117-2125.
[17] Tan M, Pang R, Le Q V. Efficientdet: Scalable and efficient object detection, Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020, 10781-10790.
[18] Dai X, Chen Y, Xiao B, et al. Dynamic head: Unifying object detection heads with attentions, Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2021, 7373-7382.
[19] Zhang H, Xu C, Zhang S. Inner-IoU: More Effective Intersection over Union Loss with Auxiliary Bounding Box, arXiv preprint arXiv. 2023,2311.02877.