IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 9 December 2023, accepted 1 January 2024, date of publication 4 January 2024,
date of current version 10 January 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3349978

RESEARCH ARTICLE

Lightweight Detection Model RM-LFPN-YOLO for Rebar Counting

HAODONG LIU, WANSHENG CHENG, CHUNWEI LI, YAOWEN XU, AND SONG FAN
School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan, Liaoning 114051, China
Corresponding author: Song Fan (fansong1983@163.com)

ABSTRACT In this study, we propose a novel lightweight detection model for rebar counting, which is rectified mobilenet lightweight feature pyramid network based on YOLO (RM-LFPN-YOLO). The model incorporates a lightweight backbone network that integrates the coordinate attention (CA) mechanism, a lightweight feature pyramid network (LFPN), and a loss function that combines focal loss and efficient intersection over union (EIOU) loss, all meticulously designed to enhance the model's performance. Experimental results demonstrate that our improved algorithm, with a mere 25.08M parameters, computes efficiently at 7.60G with an input size of 416 pixels. Additionally, it achieves an impressive average precision (AP) of 99.03% at an IOU of 0.5. The proposed lightweight model can be deployed on embedded devices and achieve efficient rebar detection and counting performance.

INDEX TERMS YOLO, attention, LFPN, focal loss.

I. INTRODUCTION

Rebar is a fundamental material extensively utilized in the steel industry, particularly in construction applications. The quantification of rebar plays a vital role in the operations of production companies, sellers, and construction sites. Recently, significant advancements in deep learning, particularly in the field of computer vision, have enabled the implementation of target detection-based counting methods for industrial inspection. But these methods often require a lot of hardware resources, a lot of network parameters, and slow forward reasoning. These factors make them unsuitable for application in real-time and resource limited embedded systems for rebar detection and counting work. Consequently, it is of great value to develop a lightweight network model that offers both fast speed and low memory consumption while ensuring accurate detection and counting of rebar targets.

Fan et al. [1] proposed a framework called CNN-DC, which first detects candidate centroids using deep CNN, then clusters the candidate centroids with distance clustering (DC), and locates the true centers of the rebars to achieve automatic rebar counting and center localization. Considering the small target of the rebar end, which is easy to miss detection, Shi et al. [2] realized the static counting of rebars by using the cascade structure of the candidate head network, and the dynamic rebar video counting on the conveyor belt was realized by using the lightweight single-scale feature map network, which not only has good real-time but also has high counting accuracy. Ghosal et al. [3] addressed the poor results of the original RetinaNet framework in dense target detection by combining a Gaussian mixture model and an expectation maximization algorithm to solve ambiguity detection and improve counting accuracy. Drawing on the SSD algorithm as well as the network feature fusion method of FPN and incorporating the RFB module, Zhou et al. [4] proposed a lightweight network for mobile devices to accomplish the rebar counting task with the advantages of high detection accuracy, a small number of model parameters, and a short training time.

Edge devices are limited by computational power, storage space, and energy efficiency when deploying large vision models [5]. These limitations have a significant impact on model performance. Computational power limitations need to be reduced by model optimization to reduce computational requirements; storage space limitations can be countered by model pruning and quantization techniques to reduce model size; and energy efficiency requires optimizing models to reduce energy consumption. Therefore, these hardware

The associate editor coordinating the review of this manuscript and approving it for publication was Gongbo Zhou.

© 2024 The Authors. This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License.
For more information, see https://creativecommons.org/licenses/by-nc-nd/4.0/

limitations need to be considered comprehensively when designing and deploying these models, and corresponding optimization techniques and innovative approaches need to be adopted to improve the performance and utility of the models on edge devices.

Therefore, in order to make the rebar counting task deployable on embedded devices, we made improvements to YOLOv4. The improved model has a significantly reduced number of parameters and computation, with only 25.08M parameters and 7.57G computation with an input size of 416 pixels, while achieving an AP value of 99.03% with an IOU of 0.5.

The main contributions of our work are as follows:

- In order to enhance the ability to perceive the spatial location information of dense steel rebar targets, coordinate attention is introduced into the inverse residual module of the lightweight backbone network for targeted optimization.
- In order to better utilize the semantic information of shallow, medium, and deep feature maps to enhance the model's ability to recognize multiscale objects, a lightweight feature fusion network, or LFPN, is designed to improve detection efficiency and accuracy.
- Aiming at the special characteristics of foreground and background in the image of the rebar dataset, focal loss is introduced into the confidence loss to balance the ratio of positive and negative samples in the rebar dataset. Aiming at the denseness characteristics of the bundled rebar targets, EIoULoss is introduced as the bounding box regression loss function to solve the problem that the bounding boxes between the adherent rebars are not easy to recognize, and to enhance the regression ability of the bounding box.

The rest of this paper is structured as follows: Section II briefly reviews the related works that are close to our method. Section III provides an elaborate account of the lightweight model network and its enhancements. Section IV elucidates the procedure for creating and augmenting the rebar dataset, along with the setup of the experimental environment. Section V carries out various experiments and subsequently analyzes their outcomes. Lastly, Section VI presents a comprehensive summary of the study.

II. RELATED WORK

With the rapid development of Convolutional Neural Networks (CNN), CNN-based target detection methods have been widely used in computer vision. In 2014, Girshick et al. [6] proposed the Regions with Convolutional Neural Network (R-CNN) method based on a CNN structure and achieved accurate detection of targets for the first time, resulting in a significant increase in the Mean Average Precision (mAP). Subsequently, improved algorithms based on candidate regions such as Fast R-CNN [7], Faster R-CNN [8], Mask R-CNN [9], and regression-based algorithms of the YOLO (You Only Look Once) series [10], [11], [12] and SSD (Single Shot MultiBox Detector) series [13], [14], [15], [16] have been successively proposed, which have improved the accuracy and real-time performance of the detection to some extent.

Due to the large model size, many parameters, and high computational complexity, it is difficult to meet the practical application requirements in terms of limited computing power, memory space, and power consumption. So many lightweight target detection algorithms have been generated, which makes it possible to deploy them on resource-constrained embedded systems. Tiny-YOLO and Tiny-SSD [17] modify the backbone network and feature enhancement network on the basis of a large model to compress the model volume, which makes the model lightweight and has high detection accuracy. In 2016, Iandola et al. [18] proposed the SqueezeNet lightweight network by drawing on the design ideas of the Inception network [19] and compressing the existing network in a lossless way, which guarantees the accuracy of the model while having a smaller number of parameters. In 2017, Howard et al. [20] first used Depthwise Separable Convolution (DSC) instead of traditional convolution and constructed the Mobilenetv1 lightweight network based on it, which reduces the number of parameters and operations while accelerating the inference of the model. In the same year, Zhang et al. [21] replaced the first 1×1 convolutional layer using group convolution based on the traditional residual units, followed by the use of channel shuffle operation on the outputs of each group so as to achieve the role of information interaction, and thus the proposed ShuffleNet v1 lightweight network is able to achieve a balance between speed and performance. In 2018, Sandler et al. [22] proposed Mobilenetv2, lightweight network based on Mobilenetv1, which introduced the inverse residual method and linear bottleneck structure through the inverse operation of ''dimension-up-convolution-decimation'', thus greatly reducing the computational amount of the depth-separable convolution in the intermediate convolution for the operation. Liang et al. [23] proposed an improved sparse R-CNN framework for traffic sign detection in autonomous vehicles. It integrates a coordinate attention block with the ResNeSt backbone network and employs a feature pyramid for multiscale detection. The method uses data augmentation to handle diverse traffic scenarios and includes novel modules like Self-Adaption Augmentation and Detection Time Augmentation for enhanced robustness. Ahmad et al. [24] proposed an enhanced version of the YOLOv1 neural network for object detection, focusing on improving detection accuracy and efficiency. Liang et al. [25] introduced the DetectFormer model, which significantly enhances object detection performance in traffic scenes for autonomous driving systems. This model improves category sensitivity and feature extraction capabilities through a Global Extract Encoder and a Category-Assisted Transformer, coupled with attention mechanisms.In this paper, the improved YOLOv4 is proposed to have a lightweight backbone and LFPN modules as the feature pyramid structure to facilitate the

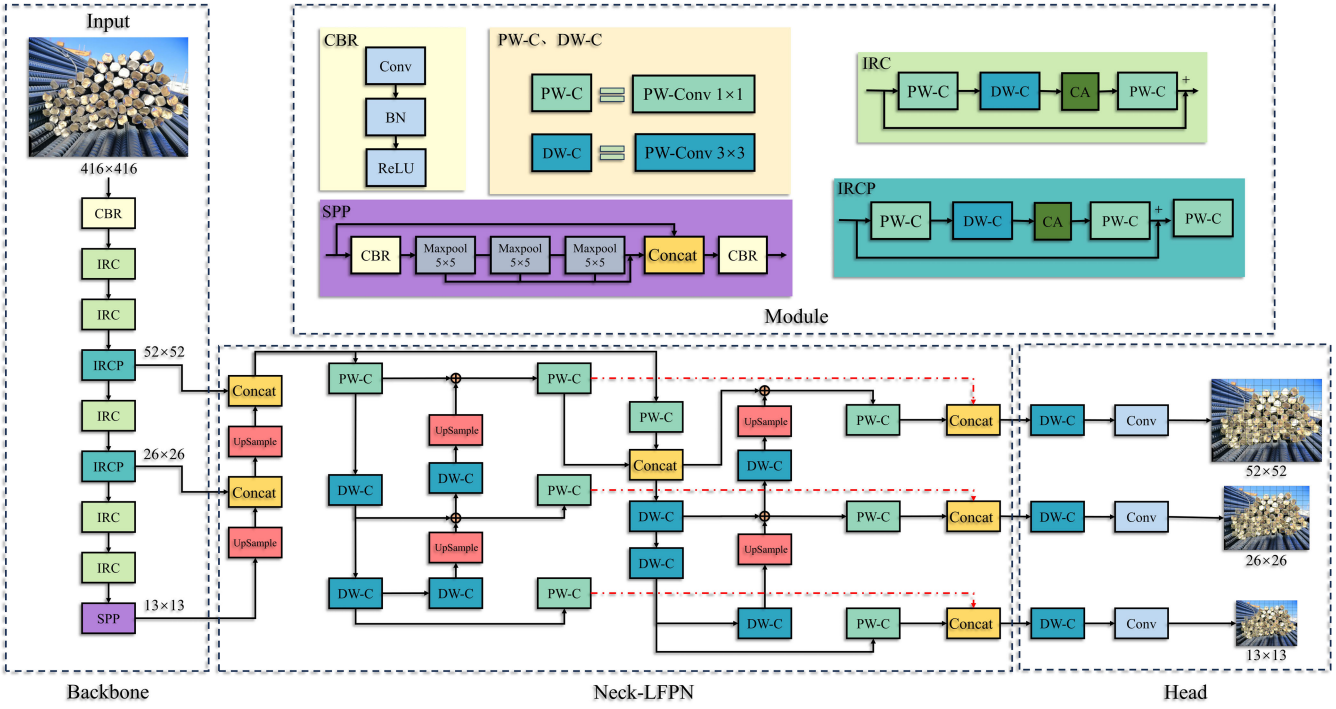


FIGURE 1. RM-LFPN-YOLO network structure.

fusion operation for multi-scale detection. The details of the RM-LFPN-YOLO model are presented in Section III.

III. METHODOLOGY

A. RM-LFPN-YOLO MODEL CONSTRUCTION BASED ON YOLOV4

YOLOv4 was chosen as the base model for our study due to its moderate structural complexity, which not only allows for efficient inspection accuracy to be maintained but also provides sufficient room for parameter tuning and optimization to be carried out to suit the specific needs of rebar inspection. This grants us the opportunity to implement lightweight improvements that can accommodate possible hardware constraints and expedite the deployment of the model.

The structure of the revised Mobilenetv2-lightweight feature pyramid network (RM-LFPN-YOLO) algorithm proposed in this paper is shown in Fig 1. The main improvements are as follows:

- An improved lightweight neural network, Mobilenetv2, was employed to extract image features. Coordinate attention (CA) was integrated into the backbone network while targeting the optimization of the inverted residual module of Mobilenetv2 to yield a new module known as IRC. Additionally, a convolution block, comprising deep and pointwise convolutions, replaced the three convolutional layers preceding and following the SPP structure. This replacement was aimed at enhancing the representation of rebar features through the lightweight Mobilenetv2 network, all at a relatively low computational cost. The layers 3 and 5 of IRC were further transformed into the module IRCP via a pointwise convolution operation, following which the results of IRCP and SPP were directed into the LFPN.
- The lightweight network model has a small amount of accuracy loss, and the detection accuracy is an important indicator of the missed and false detection situations in the rebar counting task; therefore, the network model needs to be further adjusted and optimized, and a lightweight feature fusion network structure (LFPN) is designed. The final output feature maps obtained from LFPN prediction incorporate shallow, medium, and deep features, and are thus rich in semantic information, and use deep Separable convolution, by decomposing the regular convolution into two parts: depth convolution and point convolution, which makes the LFPN structure lightweight.
- The Focal Loss function (Focal Loss) is employed to adjust the confidence loss, effectively rebalancing the ratio of positive to negative samples within the rebar dataset. Additionally, the EIOU loss replaces the initial CIOU loss for detection frame regression, enhancing localization performance when dealing with densely packed rebar targets.

B. DESIGN OF LIGHTWEIGHT BACKBONE NETWORKS

1) Mobilenetv2 OF IMPROVED BASED ON YOLO-V4

The Mobilenet series network is a lightweight neural network proposed by Google for vision applications on mobile, embedded, and other edge devices. Mobilenet mainly

employs deep separable convolution for image feature extraction, which ensures significant recognition accuracy and performance while offering the advantages of a smaller model size and faster operation speed. Mobilenetv1, Mobilenetv2, and Mobilenetv3 were tested and compared, and Mobilenetv2 was finally selected for this study. The specific experimental results are shown in Table 4.

In the application scenario of steel reinforcement detection and counting, the input images processed by our model are those of bundled steel reinforcement end faces captured from a fixed perspective. These images exhibit a high degree of density, and the spatial positional information of the steel reinforcement end faces contained within them plays a guiding role in the model's training. Hence, the lightweight neural network MobilenetV2 is initially introduced into the YOLOv4 [26] algorithm, and its integration into the backbone network is facilitated through the utilization of coordinate attention with lightweight characteristics, as shown in Fig 2b; the inverted residual module of Mobilenetv2 is subsequently optimized in a targeted manner to enhance the lightweight network while incurring a relatively low computational cost, as shown in Fig 2c. The IRC of layers 3 and 5 obtains the module IRCP through a pointwise convolution operation, as shown in Fig 2d. Mobilenetv2 representation of rebar features. The overall improved backbone process is shown in Fig 2a.

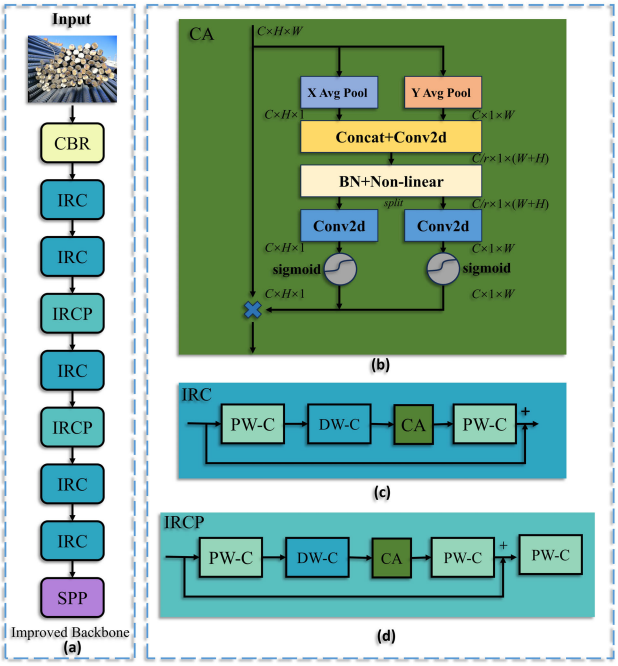


FIGURE 2. Structure of a lightweight backbone network: (a) fused CA and improved backbone network structure,(b) structure of coordinate attention, (c) optimized inverted residual module, and(d) optimized inverted residual module add pointwise convolution.

For the YOLOv4 network, a $416 \times 416 \times 3$ image is taken as input, and image features are extracted using the CSPDarknet53 backbone network. The initial effective feature layer of size $13 \times 13 \times 1024$ is obtained after three conventional convolutions, SPP structure, concatenation operations, and three more conventional convolutions. This output feature layer is then combined with two other initial effective feature layers of size $52 \times 52 \times 256$ and $26 \times 26 \times 512$, and collectively passed through the PANet (Feature Enhancement Network) to integrate information from the three feature layers, resulting in three deeper feature layers. Finally, the YOLO Head predicts the results based on the obtained feature maps.

However, for the Mobilenetv2-YOLOv4 network, the main backbone structure is obtained by removing the last three layers used for the classification task in the Mobilenetv2 network and changing the repetition count of the last bottleneck module to three. At this point, the input receives an image of size $416 \times 416 \times 3$, which first undergoes feature extraction and down-sampling operations through a ConvBNReLU module consisting of two-dimensional convolution operations, BN (Batch Normalization) layers, and ReLU6 activation functions. Subsequently, the modified Mobilenetv2's Bottleneck module is applied for further feature extraction and down-sampling, resulting in three preliminary effective layers of size $52 \times 52 \times 32$, $26 \times 26 \times 96$, and $13 \times 13 \times 320$, respectively. The $13 \times 13 \times 320$ effective feature layer is then processed by the SPP (Spatial Pyramid Pooling) structure to increase its receptive field. Afterward, the output from the SPP and the other two preliminary effective feature layers are sent to the lightweight PANet structure to accomplish the effective fusion of low-level and high-level feature information. Finally, YOLOv4's YOLO Head is responsible for predicting the target positions and class information, thereby achieving the task of counting bundled steel bars.

2) INTRODUCTION OF THE ATTENTION MECHANISM

The Coordinate Attention [27] (CA) was proposed by Hou et al. in 2021. Its essence lies in integrating coordinate information into channel attention, enabling the attention mechanism to capture long-range dependencies of spatial positions and orientations in the image. The CA module consists of two parts: coordinate information fusion and coordinate attention generation, as depicted in Fig 3b. The CA module not only captures inter-channel correlation information but also incorporates coordinate information in both horizontal and vertical directions. This encoding process enhances the model's representational capacity and improves its detection performance. In the application scenario of steel bar detection and counting, the model takes input images of bundled steel bar endfaces captured from a fixed perspective, which exhibit dense characteristics. The spatial positional information of steel bar endfaces in the input images guides the model's training. Therefore, to enhance the network's effective extraction of steel bar endface features, this paper integrates lightweight coordinate attention into the backbone network, optimizing the inverted residual modules of Mobilenetv2 with targeted improvements to significantly improve the representation of steel bar

features with relatively low computational cost. Different kinds of attentional mechanisms were tested and compared, including Squeeze-and-Excitation(SE) [28], Ghost attention(GAtt) [29], Bottleneck Attention Module(BAM) [30], Convolutional Block Attention Module(CBAM) [31], and Efficient Channel Attention(ECA) [32], and CA was finally selected for this study. The specific experimental results are shown in Table4.

C. DESIGN OF LIGHTWEIGHT FEATURE FUSION LFPN NETWORKS

In recent years, researchers have employed a multi-scale approach to address the issue of scale variation by designing efficient feature fusion structures. This allows for the enhancement of detection accuracy in various scenarios, resulting in significant achievements. Among these, the most representative is the feature pyramid structure proposed by Lin et al. [33]. In real images, the scale of the objects to be detected is approximately consistent, but their appearance features vary significantly in complexity. For conventional feature pyramid models, the feature maps used to detect objects within a specific size range are primarily constructed from a single layer or adjacent layers of the main network. Therefore, the performance of object detection for scale variation is not ideal. Considering the particularity of bundled steel bar end images, this study draws on the excellent network architecture from reference [34] and proposes an effective lightweight feature fusion network, namely the Lightweight Feature Pyramid Network (LFPN), to enhance detection efficiency and accuracy. The overall structure of LFPN, as shown in Fig 3, consists of four parts: initial feature fusion, U-shaped refined feature extraction, deep fusion, and scale feature aggregation.

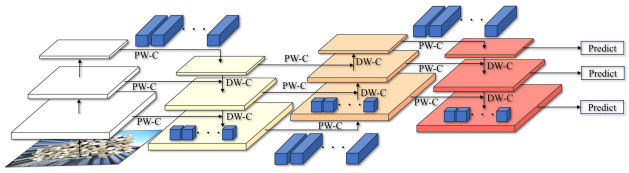


FIGURE 3. Lightweight feature pyramid network.

Compared to the traditional FPN structure, each level of the LFPN structure's feature maps used for prediction not only come from a single or adjacent layer of the backbone network but also integrate features from shallow, middle, and deep layers. Consequently, the output feature maps used for prediction in LFPN possess abundant semantic information, and the utilization of depth-wise separable convolutions renders the LFPN structure lightweight in nature.

D. OPTIMIZATION OF LOSS FUNCTIONS

1) REPLACING CIoU WITH EIOU LOSS

In the YOLOv4 algorithm, the loss function consists of three parts: regression loss in the form of CIoULoss [35], confidence loss in the form of cross-entropy loss, and classification loss. CIOU Loss, as the regression loss function, evaluates the disparity between the predicted and real frames by the area of overlap between predicted and real frames, the distance between the centers, and the ratio of width to height of the two frames. The formula is defined by Equation (1).

$$\begin{cases} v = \dfrac{4}{\pi^2}(arctan\dfrac{w^{gt}}{h^{gt}} - arctan\dfrac{w}{h}) \\ a = \dfrac{v}{(1-IOU)+v} \\ L_{\text{CIOU}} = 1 - IOU + \dfrac{d^2(b, b^{gt})}{c^2} + a \cdot v \end{cases} \tag{1}$$

where v is used to measure the similarity of the aspect ratio, a is the weight function, $a \cdot v$ represents the width-height ratio relationship between the real frame and the predicted frame, and the value of $a \cdot v$ is smaller when the two are more similar; the widths and heights of the real frame are w^{gt} and h^{gt}, respectively, and the widths and heights of the predicted frame are w and h, respectively; the centroids of the predicted frame and the real frame are b and b^{gt}, respectively, and the Euclidean distance from the centroids of the two target frames is denoted by d; the diagonal distance of the smallest rectangle capable of encompassing both the predicted and the real frames is denoted by c, IOU means Intersection over Union, which is taken to be 0.5 in this study because it is a common strategy for calculating average precison in target detection. L_{CIOU} is the loss of the CIOU for bounding box regression, as shown in Fig 4.

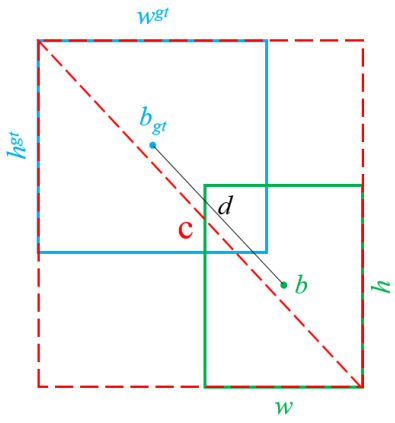


FIGURE 4. Schematic diagram of CIOU loss.

To address the shortcomings of the CIOU loss function, the EIOU loss function [36] splits the aspect ratio penalty term into the difference between the width of the predicted box and the real box and the difference between the height of the box as a penalty term, which can control the width of the two target frames. At the same time, the height of the two target frames is also consistent, so the training time of the network model is reduced and the regression accuracy is higher. The entire EIOU loss function involves three geometric factors: overlap area, center distance, and shape size. Its formula is

defined by Equation (2):

$$L_{EIOU} = 1 - IOU + \frac{\rho^2(b, b^{gt})}{c^2} + \frac{\rho^2(w, w^{gt})}{c_w^2} + \frac{\rho^2(h, h^{gt})}{c_h^2} \tag{2}$$

where the width and height of the smallest outer rectangle enclosing the two target boxes are c_w, c_h, respectively, and the Euclidean distances between the widths and heights of the true and predicted boxes are $\rho(w, w_{gt})$, $\rho(h, h_{gt})$, respectively.

2) INTEGRATING FOCAL LOSS INTO CONFIDENCE LOSS

For the target detection task, the single-stage detection algorithm based on anchor generates a large number of prediction frame samples during the network training process, in which few samples contain the target object (positive samples), and most of the samples do not contain the target's background (negative samples) and the number of positive and negative samples is very unbalanced; there are even fewer difficult samples for hard categorization and even more simple samples for easy categorization, so that the simple samples and negative samples participate in the calculation of loss in a large proportion of the network training process. The focal loss function was proposed by Tsung-Yi et al. [37] as a solution to the sample data imbalance problem of single-stage target detection algorithms by reducing the weights of negative samples and simple samples in the loss and relatively increasing the roles of positive samples and difficult samples in the training and convergence of the model.

The focal loss function balances the number of positive and negative samples by adding a weighting factor a to increase the contribution of positive samples to the loss value and relatively reduce the proportion of negative samples participating in the loss calculation, and introduces a modulation factor b to increase the contribution of difficult-to-categorize samples to the loss value and relatively reduce the proportion of easy-to-categorize samples participating in the loss calculation, thus balancing the number of difficult and easy samples. The formula for the focal loss is defined by Equation (3).

$$FL(p, y) = \begin{cases} -\alpha(1-p)^{\gamma}\log(p), & y = 1 \\ -(1-\alpha)p^{\gamma}\log(1-p), & otherwise \end{cases} \tag{3}$$

In this Equation, p represents the probability value of the predicted box containing the target, y is the true label, and $y = 1$ indicates that the sample is a positive sample. α is the weighting factor, and γ is the modulating factor. Increasing the value of α implies a greater contribution of positive samples to the loss calculation, leading to a relatively smaller value of $1 - \alpha$, indicating a reduced contribution of negative samples. This approach strengthens the model's learning ability and inclination towards positive samples. When dealing with easily classifiable positive samples (large p values), the introduction of the parameter $(1-p)^{\gamma}$ reduces their contribution to the loss calculation. Conversely, for easily classifiable negative samples (small p values), the introduction of parameter p^{γ} reduces their contribution to the loss calculation. Moreover, $\gamma > 0$ and γ enhance the model's ability to mine challenging samples, thus reinforcing the model's learning ability and inclination towards challenging samples.

The process of training the network model on the rebar dataset produces multiple prediction frames, most of which appear in regions where the rebar end faces do not exist, resulting in a large number of negative samples as a proportion of the entire sample set. At this time, applying the cross-entropy loss function as the confidence loss will make the negative samples participate in the calculation of most of the loss values, which will lead to the problem that the network model learns a large number of features of from the negative samples but lacks the useful features of the positive samples. Therefore, the focal loss function is applied to the confidence loss, and the weighting and modulation factors are introduced to reduce the contribution of the negative samples to the loss and relatively increase the contribution of the positive samples to the loss, thus balancing the ratio of the two numbers.

IV. DATASET AND IMPLEMENTATION SETTINGS

A. EXPERIMENTAL DATA

In this study, we focus on the detection and counting of steel bars in industrial automation. The techniques and implementations we have adopted are specifically designed to accurately identify and count steel bars and are not intended for identifying and counting other types of bars, such as composite bars or polymer bars. By focusing our research on rebar detection and counting, we hope to provide more accurate and reliable results. The experiments and analyses described in this paper focused on the detection and counting of steel bars and did not address other types of material.

The dataset [38] utilized in the studies is sourced from rebar manufacturers and building sites and is made available as an open-source resource. The dataset comprises a total of 250 photos that have been annotated and 200 images that have not been annotated. Each image in the collection depicts a distinct rebar end target. Furthermore, the dataset has a collection of photos depicting bundled rebars, exhibiting a diverse range of diameters spanning from 12 mm to 32 mm. Fig 5 displays a selection of example photos contained within the collection.

We made the following three assumptions while designing the model: the images are assumed to be free of severe blurring or noise, as these factors can significantly affect the accuracy of the algorithm. Good image quality is a prerequisite for improved detection accuracy; and it is assumed that all images are acquired under similar environmental conditions, e.g., during daytime or using a constant artificial light source. This helps to minimize the impact of environmental variations on the performance of the algorithm; it is assumed that the main features of the rebar, such as shape and size, are clearly visible in the images, and that these features have

FIGURE 5. Sample of some images of the dataset.

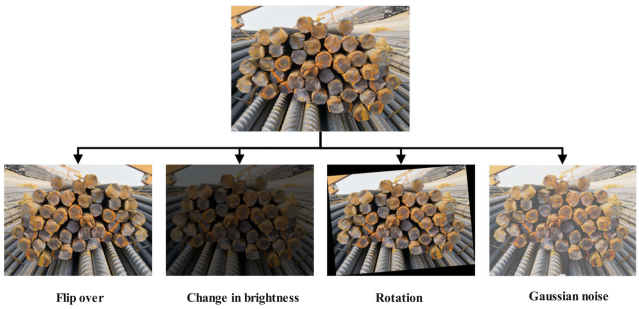


FIGURE 6. Data augmentation effect of rebar images.

TABLE 1. Dataset attributes.

Dataset	Images	Total Rebar Cross-Sections	Image Resolution
Original	450	48001	2666×2000
Augmented	1370	152516	2666×2000

some consistency from image to image. Therefore we operate through data augmentation so as to fulfill the proposed assumptions and to improve the accuracy and reliability of the algorithm in practical applications.

The process of manually annotating images with missing annotations was conducted for the purpose of this study, utilizing the LabelImg program, as depicted in Fig 6.After performing operations such as flipping, adjusting image brightness, rotating at certain angles, and adding Gaussian noise to the sample images with annotated information, the quantity and diversity of the dataset were expanded. As a result of data augmentation, a reinforced steel dataset containing 1370 images was obtained. The augmented dataset provided a comprehensive collection of 151397 steel bar cross-sections required for the experiments. During the experimental process, the dataset was partitioned into training, validation, and testing sets with a ratio of 8:1:1, each serving different purposes. As shown in Tbale 1.

B. EXPERIMENTAL ENVIRONMENT AND TRAINING PARAMETERS

To conduct the experiment, we employed the PyTorch framework [39]. The experimental hardware configuration consisted of one RTX 3090 GPU graphics card with 24GB of memory and an Intel (R) Xeon (R) Gold 6139 CPU. The software environment encompassed CUDA version 11.4. For model development, Python 3.6 was utilized, while Pycharm served as the code editor. Deep learning tasks were carried out using PyTorch 1.8.0 as the framework, and computer vision aspects were facilitated by OpenCV 4.1.2, serving as the computer vision library. The experimental environment was set up as shown in Table 2.

TABLE 2. Experimental environment.

Experimental environment	Details
Processor	Intel(R) Xeon(R) Gold 6139 CPU
Operating system	Linux
Ram	32 GB
Graphics card	RTX3090 GPU
Programming language	Python3.8
Deep learning libraries	PyTorch1.8.0
Deep learning toolkit	CUDA11.4

TABLE 3. Training parameters.

Parameter	Value
Image Size	416*416
Learning Rate	0.01
Weight Decay	0.0005
Momentum	0.937
Optimizer	SGD
Batch Size	8
Epoch	300

To ensure the fairness and effectiveness of the experiment, all fundamental parameters were kept consistent throughout the study. The dataset employed was a custom steel reinforcement dataset. During the optimization process, the momentum value was set to 0.937, and the stochastic gradient descent (SGD) algorithm was selected for optimization with a weight decay rate of 0.0005. The batch size, referring to the number of images loaded into memory at once during network training, was set to 8. We chose image size 416×416; the training process was iterated 300 times over the entire dataset before stopping, and the weight file corresponding to the lowest loss was selected as the optimal weight for the network. The training parameters are shown in Table 3.

C. EVALUATION METRICS

To assess the comprehensive performance of the model, the assessment criteria utilized in this study are precision (P), recall (R), average precision (AP), mean average precision (mAP), frames per second (FPS), and the size of the model. TP means true positive, FP means false positive, FN means false negative, and TN means true negative. The corresponding formulas are illustrated in equations (4) to (8) [40]. Precision indicates the ratio of correctly identified positive instances among the instances classified as positive,

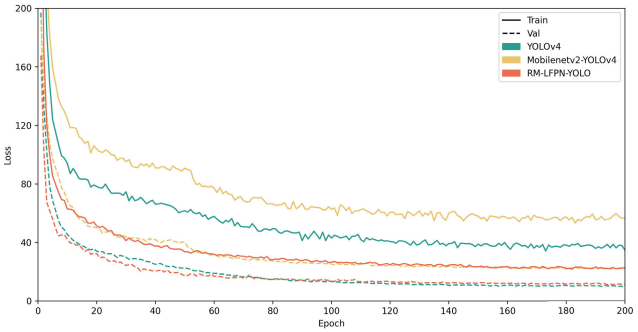


FIGURE 7. Comparison of loss curves of the improved model and the original model.

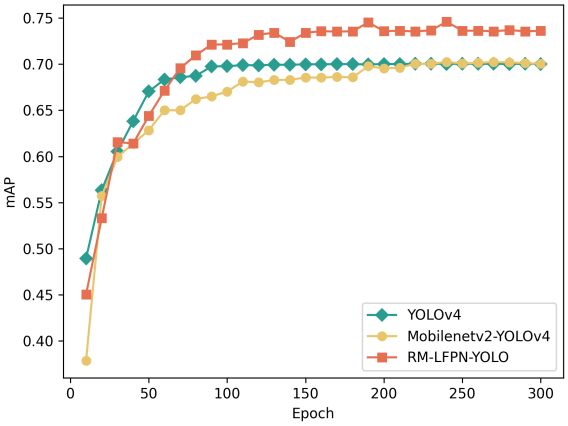


FIGURE 8. mAP curves of the training process containing the improved model.

and it is computed as follows:

$$Precision = \frac{TP}{TP + FP} \tag{4}$$

Recall signifies the fraction of true positive samples correctly identified by the model out of the total actual positive samples, and it is calculated as follows:

$$Recall = \frac{TP}{TP + FN} \tag{5}$$

The F1-Score denotes the harmonic mean of precision and recall and is computed in the following manner:

$$F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{6}$$

The formulas to compute Average Precision (AP) and Mean Average Precision (mAP) are as follows:

$$AP = \int_0^1 P(R)\, dR \tag{7}$$

$$mAP = \frac{\sum_{i=0}^{C} AP_i}{C} \tag{8}$$

In the aforementioned equation, C represents the number of all categories, AP_i denotes the AP value for class i.

V. EXPERIMENTS AND RESULTS

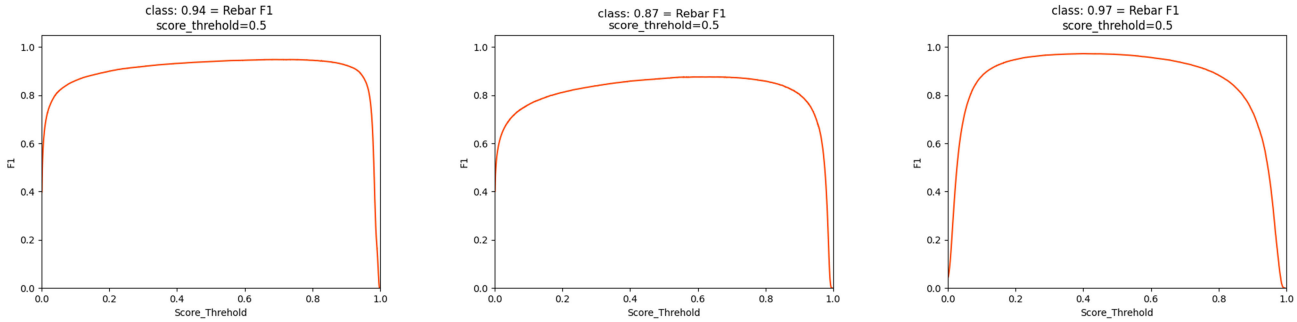
A. VALIDATION RESULTS

The RM-LFPN-YOLO network model was trained using the hardware and software environment, along with the specified basic parameter configurations and the generated dataset mentioned in the preceding section. As illustrated in Fig 7, the loss curves of the rebar validation set for the YOLOv4, Mobilenetv2-YOLOv4, and RM-LFPN-YOLO network models are presented. Additionally, Fig 8 displays the mAP curves during the training of the three network models.

From the figure, it is evident that the loss values exhibit an overall decreasing trend and stabilize rapidly. The RM-LFPN-YOLO model demonstrates faster convergence speed, a lower loss value, and a higher mAP compared to the other two models, thus showcasing superior learning efficacy. Fig 9 illustrates the F1-score curves of the three network models, and the RM-LFPN-YOLO model achieves a higher F1-score within the threshold range of 0.2-0.8. Fig 10 displays the P-R curves of the three network models, and the P-R curve of the RM-LFPN-YOLO model aligns closer to the upper right, indicating a greater average accuracy. Consequently, the RM-LFPN-YOLO model exhibits superior average accuracy, thereby resulting in enhanced detection performance.

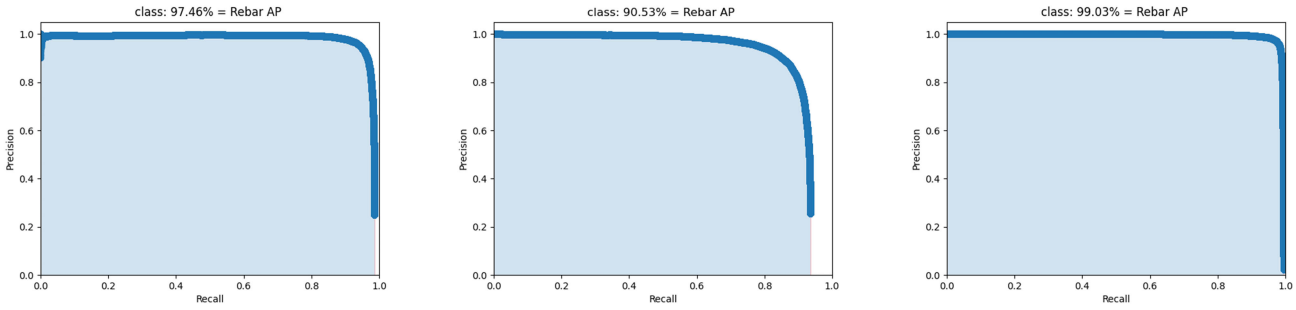
In this experiment, the model underwent evaluation using a set of quantitative metrics identical to those employed in the preceding section. The pertinent evaluation outcomes are meticulously presented in Table. 4.

As can be observed from Table4, within the same experimental environment and training dataset, the RM-LFPN-YOLO network model proposed in this study exhibits significant improvements in detection accuracy. Specifically, the AP value shows an increase of 0.53 and 8.5 percentage points when compared to YOLOv4 and Mobilenet-YOLOv4, respectively. Moreover, the RM-LFPN-YOLO model achieves the highest F1-score of 0.97, surpassing both YOLOv4 and Mobilenet-YOLOv4 in this aspect. Regarding model complexity, RM-LFPN-YOLO demonstrates notable advantages. It possesses a significantly smaller number of parameters, totaling only 25.08MB. This value is 16.12MB less than the parameter count of Mobilenet-YOLOv4. Furthermore, RM-LFPN-YOLO's computation volume is the smallest, measuring only 7.57G.With respect to detection speed, RM-LFPN-YOLO achieves a frames per second (FPS) of 77.73, which is 19.35 FPS higher than YOLOv4, but lower than Mobilenet-YOLOv4. Interestingly, the experiments reveal that the addition of the CA enhances detection accuracy but negatively impacts the detection speed of the model. However, after the removal of the CA attention mechanism, the experimentally measured FPS notably increased to 80.23.Here we are faced with a classic trade-off between accuracy and speed. In many cases, it is a reasonable choice to reduce the speed slightly to obtain high-accuracy. Especially in some application scenarios where high accuracy detection is required, this trade-off is acceptable.



(a) YOLOv4 **(b)** Mobilenetv2-YOLOv4 **(c)** RM-LFPN-YOLO

FIGURE 9. F1-score: (a) YOLOv4, (b) Mobilenetv2-YOLOv4, (c) RM-LFPN-YOLO.



(a) YOLOv4 **(b)** Mobilenetv2-YOLOv4 **(c)** RM-LFPN-YOLO

FIGURE 10. P-R: (a) YOLOv4, (b) Mobilenetv2-YOLOv4, (c) RM-LFPN-YOLO.

TABLE 4. Comparison of performance metrics across models.

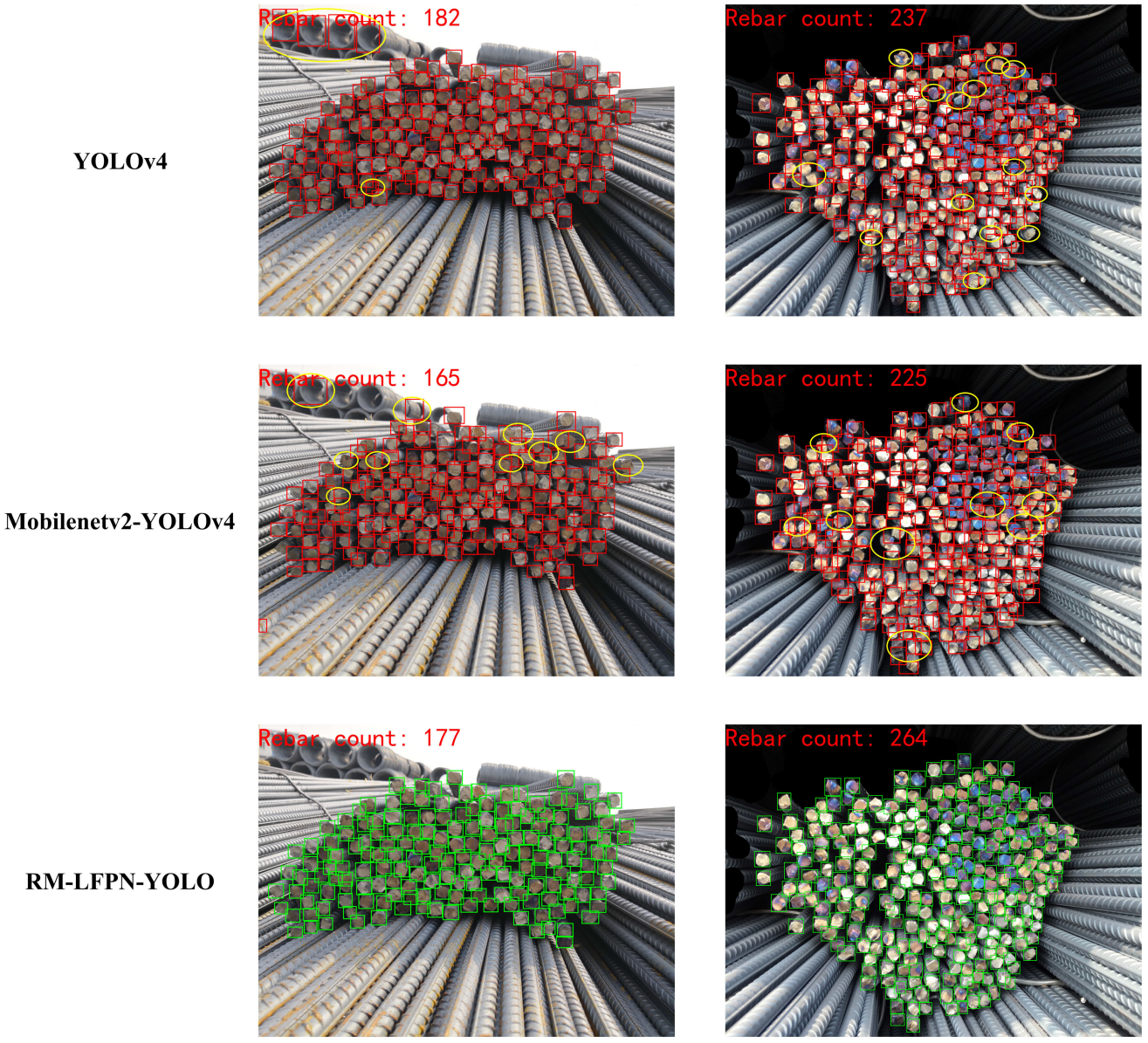
Model	AP(%)	Params(MB)	Flops(G)	F1-score	FPS
YOLOv4	97.46	245.53	60.53	0.94	58.38
Mobilenetv1-YOLOv4	88.11	48.42	10.65	0.85	69.42
Mobilenetv2-YOLOv4	**90.53**	**41.20**	**8.28**	**0.87**	**78.26**
Mobilenetv3-YOLOv4	85.21	44.74	7.71	0.82	64.71
RM-LFPN-YOLO(SE)	98.23	25.00	7.75	0.97	66.56
RM-LFPN-YOLO(GAtt)	98.49	25.55	7.67	0.97	63.92
RM-LFPN-YOLO(BAM)	98.62	25.28	7.58	0.96	63.85
RM-LFPN-YOLO(CBAM)	98.78	25.13	7.57	0.97	67.50
RM-LFPN-YOLO(ECA)	98.91	25.00	7.57	0.96	70.49
RM-LFPN-YOLO(ours)	**99.03**	**25.08**	**7.57**	**0.97**	**77.73**

B. TESTING RESULTS

In our study's implementation, to enhance the accuracy and reliability of our method, we employed a dual-sided counting strategy. This approach involves simultaneously counting both sides of the rebar bundle and cross-referencing the results. In instances where there are disparities in the counts, we conducted manual verifications to ensure data accuracy. For the counting of concealed reinforcement, we treated the respective area as if it contained no reinforcement.

Based on the analysis of experimental data, the enhanced network model exhibits attributes of superior detection accuracy, low complexity, and rapid detection speed. Additionally, it effectively addresses issues of missed detection caused by target adhesion or complex end face features, along with the problem of misdetection for non-steel end faces with analogous characteristics. Consequently, it yields a more favorable detection outcome.

Fig 11 illustrates a comparative analysis of the detection outcomes achieved by YOLOv4, Mobilenetv2-YOLOv4, and RM-LFPN-YOLO when applied to predict rebar end face images. The red and green bounding boxes correspond to the detection outputs of the original YOLOv4 and the enhanced algorithm, respectively, with the dissimilarities between them indicated by encirclement in yellow. Notably, RM-LFPN-YOLO consistently demonstrates superior performance in various real-world scenarios concerning rebar images.



(a) Yolov4 and Mobilenetv2-Yolov4 have false detections, while RM-LFPN-YOLO does not

(b) Yolov4 and Mobilenetv2-Yolov4 have miss detections, while RM-LFPN-YOLO does not

FIGURE 11. Comparison of the detection effect between the original network model and the improved network model.

C. COMPARISON WITH OTHER DETECTORS

The experimental evaluation criteria are consistent with those described in the preceding section, and the detailed evaluation outcomes are presented in Table. 5.

As can be observed from Table5, within the equivalent experimental environment and training dataset, the proposed RM-LFPN-YOLO network model in this study demonstrates superior AP value concerning detection accuracy compared to the majority of conventional algorithms. Furthermore, it significantly reduces model complexity and computational requirements while maintaining a better detection speed than the original algorithm. These findings substantiate the experiment's significance and practicality.

D. TESTED ON OTHER DATASETS

We chose another rebar dataset [41] from "UKPetra Huawei Certified ICT Associate AI Track 2021" because it is highly relevant to our research area. We use the previously trained model to test it directly on the new dataset without re-training

TABLE 5. Comparison of evaluation indicators of classical models.

Model	AP(%)	Params(MB)	Flops(G)	F1-score	FPS
FAST-RCNN	71.27	522.99	370.21	0.73	15.06
SSD	76.57	100.27	62.74	0.88	92.07
YOLOv3	99.23	236.32	66.17	0.98	68.86
YOLOv5	99.06	46.56	109.58	0.98	67.69
YOLOX	98.71	54.21	156.01	0.98	67.69
YOLOv7	99.13	37.62	106.47	0.98	26.26
YOLOv8	97.09	11.16	28.81	0.77	62.31
RM-LFPN-YOLO(ours)	**99.03**	**25.08**	**7.57**	**0.97**	**77.73**

TABLE 6. Tested on other datasets.

Model	AP(%)	FPS
YOLOv4	98.79	57.41
Mobilenetv2-YOLOv4	88.44	79.26
RM-LFPN-YOLO(ours)	**98.94**	**76.26**

it on the new dataset, and experimentally verify that the same good detection accuracy can be achieved, which proves that our previously trained model has good generalization on the rebar detection task. A comparison of the specific experimental results is shown in Table. 6.

VI. CONCLUSION

This study introduces a novel lightweight model for automatic rebar counting. The model integrates coordinate attention with Mobilenetv2, improving spatial recognition of dense rebars, and features a new lightweight feature fusion network, LFPN. Advanced loss functions like focal loss and EIoULoss-based regression loss are employed, enhancing bounding box accuracy. Compared to Mobilenetv2-YOLOv4, the newly constructed network model shows an improvement of 8.5 percentage points in Average Precision (AP), while reducing the number of parameters by 39.1%. In comparison to YOLOv4, it achieves a 1.57 percentage point increase in AP with significant reductions in both parameter and computational requirements, resulting in a 19.35 frames per second (FPS) improvement.

In future research endeavors, we aim to enhance our methodology by incorporating object instance segmentation methods specifically tailored for rebar end face analysis. We anticipate that this refined approach will significantly improve accuracy, particularly in scenarios where rebar end faces are heavily occluded. Such an advancement is expected to surpass the efficacy of conventional object detection-based algorithms.

REFERENCES

[1] Z. Fan, J. Lu, B. Qiu, T. Jiang, K. An, A. N. J. Raj, and C. Wei, "Automated steel bar counting and center localization with convolutional neural networks," 2019, *arXiv:1906.00891*.
[2] J. Shi, X. Jiang, and C. Guillemot, "A framework for learning depth from a flexible subset of dense and sparse light field views," *IEEE Trans. Image Process.*, vol. 28, no. 12, pp. 5867–5880, Dec. 2019.
[3] S. Ghosal, B. Zheng, S. C. Chapman, A. B. Potgieter, D. R. Jordan, X. Wang, A. K. Singh, A. Singh, M. Hirafuji, S. Ninomiya, B. Ganapathysubramanian, S. Sarkar, and W. Guo, "A weakly supervised deep learning framework for sorghum head detection and counting," *Plant Phenomics*, vol. 2019, pp. 1–14, Jan. 2019.
[4] Q. Zhou, Z. Qu, and F.-R. Ju, "A lightweight network for crack detection with split exchange convolution and multi-scale features fusion," *IEEE Trans. Intell. Vehicles*, vol. 8, no. 3, pp. 2296–2306, Mar. 2023.
[5] J. Chen and X. Ran, "Deep learning with edge computing: A review," *Proc. IEEE*, vol. 107, no. 8, pp. 1655–1674, Aug. 2019.
[6] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Region-based convolutional networks for accurate object detection and segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 1, pp. 142–158, Jan. 2016.
[7] S. Ren, K. He, R. Girshick, X. Zhang, and J. Sun, "Object detection networks on convolutional feature maps," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 7, pp. 1476–1481, Jul. 2017.
[8] X. Xu, M. Zhao, P. Shi, R. Ren, X. He, X. Wei, and H. Yang, "Crack detection and comparison study based on faster R-CNN and mask R-CNN," *Sensors*, vol. 22, no. 3, p. 1215, Feb. 2022.
[9] X. Bi, J. Hu, B. Xiao, W. Li, and X. Gao, "IEMask R-CNN: Information-enhanced mask R-CNN," *IEEE Trans. Big Data*, vol. 9, no. 2, pp. 688–700, Apr. 2023.
[10] W. Fang, L. Wang, and P. Ren, "Tinier-YOLO: A real-time object detection method for constrained environments," *IEEE Access*, vol. 8, pp. 1935–1944, 2020.
[11] M. D. O. Barreiros, D. D. O. Dantas, L. C. D. O. Silva, S. Ribeiro, and A. K. Barros, "Zebrafish tracking using YOLOv2 and Kalman filter," *Sci. Rep.*, vol. 11, no. 1, p. 3219, Feb. 2021.
[12] M. O. Lawal, "Tomato detection based on modified YOLOv3 framework," *Sci. Rep.*, vol. 11, no. 1, p. 1447, Jan. 2021.
[13] F.-L. Xu, Y.-L. Li, Y. Wang, W. He, X.-Z. Kong, N. Qin, W.-X. Liu, W.-J. Wu, and S. E. Jorgensen, "Key issues for the development and application of the species sensitivity distribution (SSD) model for ecological risk assessment," *Ecological Indicators*, vol. 54, pp. 227–237, Jul. 2015.
[14] B. Mao, S. Wu, and L. Duan, "Improving the SSD performance by exploiting request characteristics and internal parallelism," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 37, no. 2, pp. 472–484, Feb. 2018.
[15] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Y. Fu, and A. C. Berg, "SSD: Single shot MultiBox detector," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2016, pp. 11–14.
[16] Z. Shen, Z. Liu, J. Li, Y.-G. Jiang, Y. Chen, and X. Xue, "Object detection from scratch with deep supervision," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 2, pp. 398–412, Feb. 2020.
[17] A. Wong, M. Famouri, and M. J. Shafiee, "AttendNets: Tiny deep image recognition neural networks for the edge via visual attention condensers," 2020, *arXiv:2009.14385*.
[18] F. N. Iandola, M. W. Moskewicz, K. Ashraf, S. Han, W. J. Dally, and K. Keutzer, "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <1 MB model size," 2016, *arXiv:1602.07360*.
[19] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2818–2826.
[20] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient convolutional neural networks for mobile vision applications," 2017, *arXiv:1704.04861*.
[21] X. Zhang, X. Zhou, M. Lin, and J. Sun, "ShuffleNet: An extremely efficient convolutional neural network for mobile devices," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6848–6856.
[22] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4510–4520.

[23] T. Liang, H. Bao, W. Pan, and F. Pan, "Traffic sign detection via improved sparse R-CNN for autonomous vehicles," *J. Adv. Transp.*, vol. 2022, pp. 1–16, Mar. 2022.
[24] T. Ahmad, Y. Ma, M. Yahya, B. Ahmad, S. Nazir, and A. U. Haq, "Object detection through modified Yolo neural network," *Scientific Program.*, vol. 2020, pp. 1–10, Jun. 2020.
[25] T. Liang, H. Bao, W. Pan, X. Fan, and H. Li, "DetectFormer: Category-assisted transformer for traffic scene object detection," *Sensors*, vol. 22, no. 13, p. 4833, Jun. 2022.
[26] A. Bochkovskiy, C. Y. Wang, and H. Y. M. Liao, "YOLOv4: Optimal speed and accuracy of object detection," 2020, *arXiv:2004.10934*.
[27] Q. Hou, D. Zhou, and J. Feng, "Coordinate attention for efficient mobile network design," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 13708–13717.
[28] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," 2018, *arXiv:1709.01507*.
[29] H. Touvron et al., "Llama 2: Open foundation and fine-tuned chat models," 2023, *arXiv:2307.09288*.
[30] J. Park, S. Woo, J. Y. Lee, and I. S. Kweon, "BAM: Bottleneck attention module," 2018, *arXiv:1807.06514*.
[31] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "CBAM: Convolutional block attention module," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 3–19.
[32] Q. Wang, B. Wu, P. Zhu, P. Li, W. Zuo, and Q. Hu, "ECA-Net: Efficient channel attention for deep convolutional neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 11531–11539.
[33] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 936–944.
[34] C. Deng, M. Wang, L. Liu, Y. Liu, and Y. Jiang, "Extended feature pyramid network for small object detection," *IEEE Trans. Multimedia*, vol. 24, pp. 1968–1979, 2022.
[35] J. Gao, Y. Chen, Y. Wei, and J. Li, "Detection of specific building in remote sensing images using a novel YOLO-S-CIOU model. Case: Gas station identification," *Sensors*, vol. 21, no. 4, p. 1375, Feb. 2021.
[36] H. Peng and S. Yu, "A systematic IoU-related method: Beyond simplified regression for better localization," *IEEE Trans. Image Process.*, vol. 30, pp. 5032–5044, 2021.
[37] Y.-F. Zhang, W. Ren, Z. Zhang, Z. Jia, L. Wang, and T. Tan, "Focal and efficient IOU loss for accurate bounding box regression," *Neurocomputing*, vol. 506, pp. 146–157, Sep. 2022.
[38] H. Zhang. *DCIC Rebar Quantity AI Recognition Baseline 0.98+*. Accessed: Apr. 2023. [Online]. Available: https://github.com/HarleysZhang/detect_steel_number
[39] K. M. Chen, E. M. Cofer, J. Zhou, and O. G. Troyanskaya, "Selene: A PyTorch-based deep learning library for sequence data," *Nature Methods*, vol. 16, no. 4, pp. 315–318, Apr. 2019.
[40] K. Xia, Z. Lv, K. Liu, Z. Lu, C. Zhou, H. Zhu, and X. Chen, "Global contextual attention augmented YOLO with ConvMixer prediction heads for PCB surface defect detection," *Sci. Rep.*, vol. 13, no. 1, p. 9805, Jun. 2023.
[41] N. Veron, "Rebar counting computer vision using faster RCNN with ResNet-50 pretrained backbone," Huawei Certified ICT Associate Competition, AI Track Indonesia (UKPetra), Oct. 2021. [Online]. Available: https://github.com/illegallyCrushed/UKPetra-Huawei-Certified-ICT-Associate-AI-Track-2021#datasets

HAODONG LIU received the bachelor's degree in automation from the Mingde College, Northwestern Polytechnical University. He is currently pursuing the master's degree in electronic information with the University of Science and Technology Liaoning. His research interests include image processing on deep learning and intelligent control.

WANSHENG CHENG received the Ph.D. degree from the School of Mechatronics Engineering, Harbin Institute of Technology, Harbin, China, in 2008. He is currently the Director of the Intelligent Manufacturing Engineering Technology Center, University of Science and Technology Liaoning. His current research interests include signal processing, intelligent equipment, and robot application. He is a member of the Robotics Committee of the Chinese Association of Automation.

CHUNWEI LI received the bachelor's degree in automation from the Qingdao University of Technology. He is currently pursuing the master's degree in electronic information with the University of Science and Technology Liaoning. His research interests include the direction of deep learning for target detection and industrial recognition.

YAOWEN XU received the bachelor's degree in information and computing science from Shenyang Aerospace University. He is currently pursuing the master's degree in electronic information with the University of Science and Technology Liaoning. His research interests include data mining and deep learning.

SONG FAN received the Ph.D. degree in control theory and control engineering from Northeastern University, Shenyang, China, in 2020. He is currently with the School of Electronic and Information Engineering, University of Science and Technology Liaoning. His current research interests include data-driven fault diagnosis, machine learning, and machine vision. He is a member of the China Computer Federation.