POWER FIELD OPERATORS DETECTION ALGORITHM BASED ON SCR-YOLOv8s

Lingwen MENG[1]*, Guanghui XI[1], Siwu YU[1], Siqi GUO[1], Zhonghai RUAN[2]

To address undetected instances and false alarms in YOLOv8s caused by small and occluded targets in power field operator detection, an improved algorithm, Sparse Depthwise Convolution and CARAFE - YOLOv8s(SCR-YOLOv8s), based on Sparse Depthwise Convolution (SPDConv) and CARAFE, is proposed. First, the Sparse Depthwise Convolution and GAM – C2f (SGAM-C2f) module is designed, replacing some backbone convolutions with non-strided convolution Sparse Depthwise Convolution (SPDConv) while introducing the Global Attention Mechanism (GAM) to enhance global contextual feature extraction for small targets. Second, the lightweight upsampling operator CARAFE replaces the original method to expand the receptive field. Third, the Repulsion-Loss function is integrated to improve detection accuracy for occluded targets in complex environments. Finally, the dynamic detection head DyHead is adopted to enhance small target processing. Experiments show SCR-YOLOv8s achieves 90.23% mAP@0.5, a 4.81% improvement over YOLOv8s, with 15.05M parameters and 32.9GFLOPs, meeting engineering application requirements.

Keywords: deep learning, SPDConv, SCR-YOLOv8s, global attention mechanism, Dynamic Head

1. Introduction

With the rapid development of the power industry, enhancing the safety and efficiency of field operations through real-time image analysis and fast, accurate operator detection has become crucial. This approach not only boosts operational efficiency but also reduces risks and costs associated with manual inspections, providing immediate feedback to ensure grid stability and safety. Traditional manual inspections are inefficient and less accurate, highlighting the need for advanced detection algorithms. Recently, artificial intelligence, particularly deep learning, has been applied for operator detection in this sector. Object detection algorithms are primarily divided into two paradigms: two-stage and single-stage. The former, including Fast Region-based Convolutional Neural Network (R-CNN) [1] and Faster R-CNN [2], is renowned for its high detection accuracy. The latter, which includes models like Single Shot MultiBox Detector (SSD) [3] and the You Only Look Once (YOLO) series [4], is celebrated for its exceptional inference speed. As technology evolves, single-stage algorithms have improved in accuracy while maintaining speed, making them ideal for the power sector's demands for

[1] Electric Power Research Institute of Guizhou Power Grid Co. Ltd, Guiyang, Guizhou, China, 550002, *Corresponding author's e-mail: menglw000@163.com
[2] GuangZhou Power Electrical Technology Co., Ltd., Guangzhou, Guangdong, China, 510535

quick response and stability due to their simpler, faster structure.

Among single-stage detection algorithms, YOLO series models have been extensively used for detecting field operators in power operation scenarios. Jia et al. proposed SDS-YOLOv8-tiny, a lightweight YOLO-based model for helmet detection in complex power construction scenarios, which leverages the LAMP pruning algorithm to reduce redundant parameters and computation and incorporates the novel logic-feature dual-fusion BCCKD distillation strategy to enhance contextual semantic modeling [5]. Liu et al. [6] proposed FD-YOLO, an enhanced YOLO-based model that modifies the Backbone, Neck, and Head structures while adopting an anchor-free bounding box selection strategy. By integrating the C-JDE model, the framework achieved higher detection accuracy for power workers operating in confined spaces, demonstrating significant gains in average precision. In aquaculture research, an advanced detection algorithm named DDEYOLOv9 was proposed by Li et al. [7], building upon the YOLOv9 framework. To address the challenge of abnormal fish behavior recognition, they constructed a specialized dataset focusing on "Takifugu rubripes" and integrated several enhancements, including DRNELAN4, DCNv4-DyHead, and EMA-SlideLoss, to optimize model performance.

Although the aforementioned YOLO-based algorithms for detecting field operators in power operations have optimized model detection performance in several aspects, their accuracy remains inadequate when handling small and occluded targets. Furthermore, many existing models fail to satisfy the critical requirements of real-time processing and compact architecture essential for power industry applications, primarily due to excessive computational complexity and suboptimal inference speeds. Currently, YOLOv8s exhibits the best detection performance among similar algorithms. In light of this, this paper proposes SCR-YOLOv8s (SGAM_C2f-CARAFE-Repulsion YOLOv8s), an enhanced YOLOv8s variant for power field operator detection. The SCR acronym denotes three key innovations: (1)S-SGAM_C2f: the design of a new module SPDConv-based Global Attention Mechanism with Cross Stage Partial fast (hereafter denoted as SGAM_C2f) based on improvements to the C2f structure, in which SPDConv replaces some convolutional layers and the GAM attention mechanism is incorporated, aiming to enhance global contextual feature extraction and improve the Backbone's capability in capturing small target details; (2)C-CARAFE, the integration of Content-Aware ReAssembly of Features, which we denote as CARAFE , a lightweight upsampling technique, to effectively enlarge the model's receptive field; (3)R- Repulsion-Loss, the adoption of the Repulsion-Loss function to increase detection accuracy for occluded targets in complex power operation settings; the implementation of DyHead as a dynamic detection head supersedes conventional architectures, significantly enhancing model adaptability and small-target detection capabilities in complex environments.

2. SCR-YOLOv8s algorithm

2.1 YOLOv8s

The YOLOv8s model is structured into three parts: Backbone, Neck, and Head, optimized for performance and lightweight design. The Backbone features an enhanced CSPDarkNet [8] with C3 and ELAN [9] modules plus the novel Cross Stage Partial fast (hereafter denoted as C2f) module for improved information flow, culminating in an SPPF module for multi-scale feature fusion. The Neck employs a simplified Pixel Aggregation Network (PAN) [10] architecture for efficient shallow-deep information integration, while the Head splits into two branches for object classification and bounding box regression using Generalized Focal Loss (GFL) [11] and CIoU [12]. Despite its advancements, YOLOv8s struggles with underwater and small object detection due to limitations in feature extraction, upsampling methods lacking semantic consideration, insufficient localization accuracy for small objects, and challenges in multi-scale information processing.

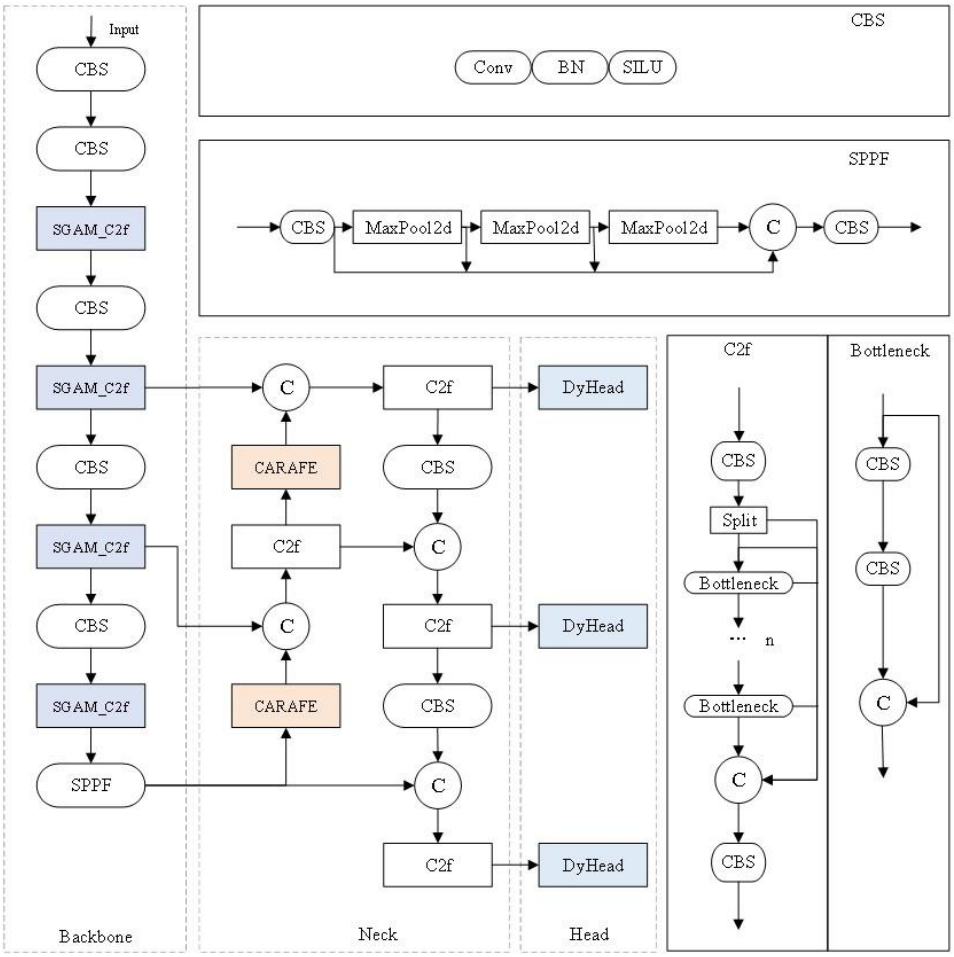

Fig. 1. SCR-YOLOv8s algorithm structure

To address these constraints, our work implements four key modifications to the YOLOv8s framework: (1) development of the SGAM_C2f component to augment feature extraction for diminutive objects; (2) replacement of conventional upsampling with the computationally efficient CARAFE mechanism to broaden perceptual scope; (3) incorporation of Repulsion-Loss to enhance precision when detecting obscured items in intricate electrical infrastructure scenarios; and (4) deployment of the DyHead dynamic detection module to supersede the standard head configuration, substantially improving identification of small-scale targets in demanding conditions. The architectural configuration of our enhanced SCR-YOLOv8s system is depicted in Fig. 1.

2.2 SGAM_C2f structure

2.2.1 Global attention mechanism

The GAM (Global Attention Mechanism) enhances the CBAM (Convolutional Block Attention Module) by reconfiguring its submodules, such as Channel Attention and Spatial Attention. An overview of the GAM structure is provided in Fig. 2.

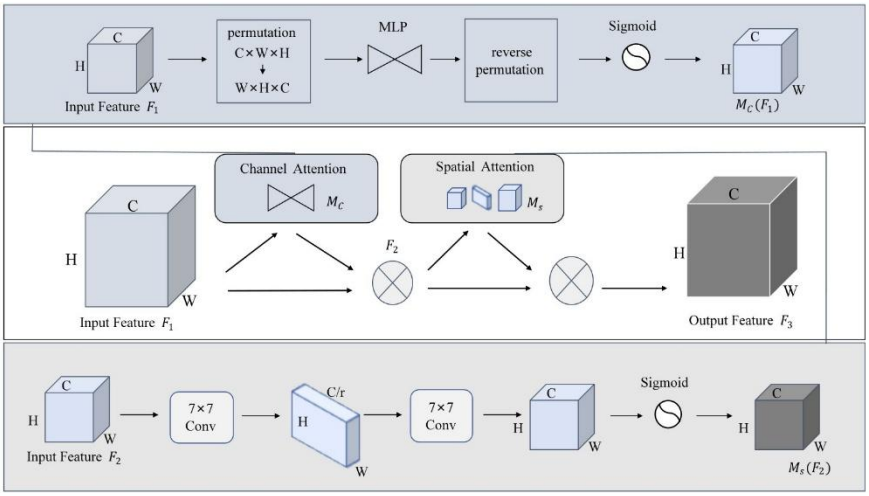

Fig. 2. GAM structure diagram

The GAM receives a map F_1 with dimension $\mathrm{C}\times\mathrm{W}\times\mathrm{H}$, C denotes the number of channels, W and H represent the width and height of the feature map. For F_1, The channel attention component employs a 3D axis permutation operation to maintain spatial structural integrity, yielding a dimensionally transformed feature tensor $\mathrm{W}\times\mathrm{H}\times\mathrm{C}$. This information is then processed through a Multilayer Perceptron (MLP) to strengthen the inter-dimensional relationships between channels and spatial dimensions. Subsequently, the feature map is reverted to its original dimensions using inverse permutation and the Sigmoid activation function,

resulting in the output from the Channel Attention module.

The channel-refined feature tensor $M_C(F_1)$, generated by the Channel Attention mechanism, undergoes Hadamard product operation with the initial input features F_1, producing an enhanced representation F_2 which serves as input to the Spatial Attention component. This intermediate feature set F_2 then sequentially processes through a pair of 7×7 convolutional operators for spatial context aggregation, followed by Sigmoid normalization to yield the spatially-weighted feature matrix $M_S(F_2)$.

The final GAM output F_3 is computed as the element-wise multiplicative combination of the intermediate feature map F_2 and the spatially-processed features $M_S(F_2)$.

2.2.2 Non-stride convolution SPDConv

The CBS module in the YOLOv8s Backbone network contains several strided convolutions and pooling layers. During the feature extraction phase, fine-grained target details can be lost, leading to subpar performance in detecting low-resolution images and small objects. To address this issue, a non-stride convolution called SPDConv is incorporated to optimize the C2f architectural component in the backbone network. The operational principle of SPDConv is depicted in Fig. 3.

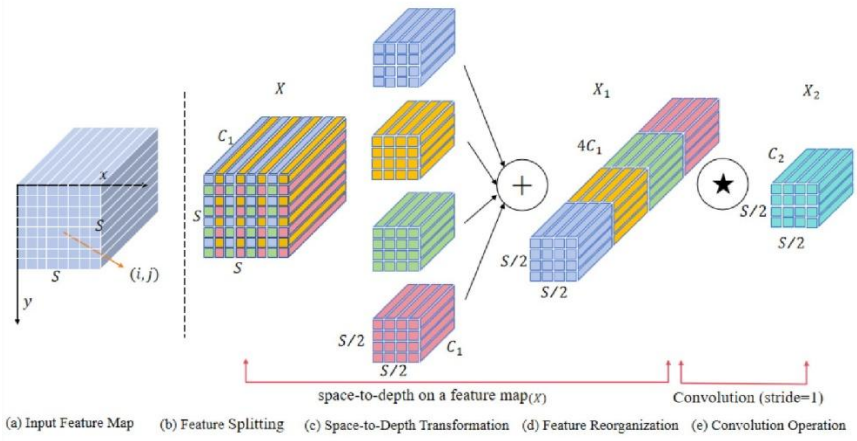

Fig. 3. Illustration of SPD-Conv when $scale = 2$

SPDConv consists of two parts: the space-to-depth layer and the non-stride convolution layer (NS-Conv). For an input feature map X of dimension $S \times S \times C_1$, the space-to-depth layer splits it into a series of sub-feature maps $f_{x,y}$, each of

dimension $S/scale \times S/scale \times C_1$, which are then concatenated along the channel dimension to form a feature map X_1, with a new dimension of $S/scale \times S/scale \times (2 \cdot scale \cdot C_1)$, The non-stride convolution layer performs convolution with kernel size 1×1 on the input feature map X_1 to reduce its dimensionality while preserving as much discriminative information as possible, resulting in the final SPDConv output feature map X_2 of dimension $S/scale \times S/scale \times C_2$ Compared to traditional convolution operations, SPDConv retains more information, making it more suitable for small target detection.

2.2.3 SGAM_C2f structure

To optimize YOLOv8s' detection capability in electrical utility environments, particularly for small-scale objects, we develop the novel SGAM_C2f architecture. This redesign incorporates: (1) integration of the Global Attention Mechanism (GAM) into the C2f module's Bottleneck components, (2) strategic replacement of conventional CBS modules with SPDConv operations, and (3) complete structural reorganization. The resulting architecture is illustrated in Fig. 4.

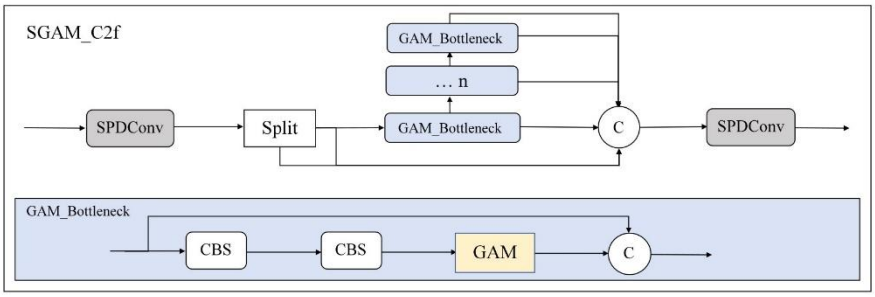

Fig. 4. SGAM_C2f structure diagram

In the SGAM_C2f structure, SPDConv replaces the original CBS convolution module, and the Bottleneck is improved by using GAM_Bottleneck to replace the original Bottleneck in C2f. The GAM_Bottleneck module initially processes input features through a dual-convolutional layer sequence, subsequently applying GAM operations. The attention-enhanced features are then additively combined with the original input through residual connection, generating the module's final output representation. Compared to C2f, SGAM_C2f captures richer detailed features of the feature map, enhances global contextual information, and strengthens cross-dimensional channel-space correlations. This makes the model more focused on the target’s intrinsic features while reducing reliance on redundant information generated during the model iteration process, rendering it specifically

effective for small-object detection applications.

2.3 Lightweight upsampling operator CARAFE

In object detection, typical upsampling techniques encompass interpolation methods like nearest-neighbor interpolation, bilinear interpolation [13], and bicubic interpolation [14]. YOLOv8s uses nearest neighbor interpolation to perform upsampling on high-level features. However, nearest neighbor interpolation relies solely on the spatial position of the feature map to determine the upsampling kernel, without taking into account the underlying semantic information. Moreover, its limited receptive field is insufficient for capturing detailed semantic features needed in dense prediction tasks. To address this, this paper adopts CARAFE[15], a lightweight and efficient upsampling operator that performs feature map upsampling through content-aware reassembly. The architecture of CARAFE is presented in Fig. 5.

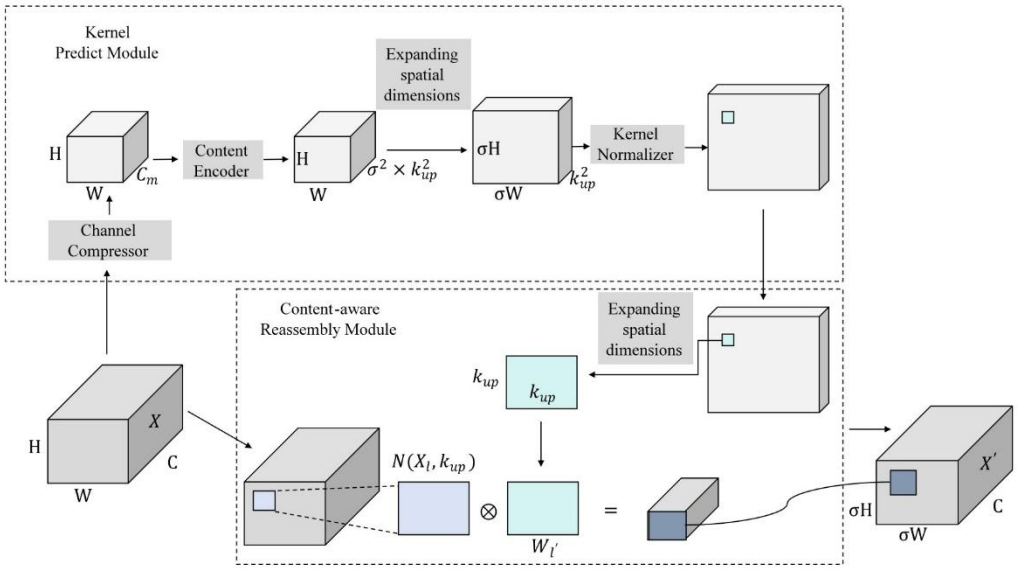

Fig. 5. CAREFE structural diagram

CARAFE is primarily composed of two components: a dynamic kernel generation unit and a feature reorganization module with content adaptation capabilities. The Kernel Prediction Module includes three subparts: Channel Compressor, Content Encoder, and Kernel Normalizer. When processing an input feature tensor of dimensions D×D×D, the channel reducer initially performs dimensionality reduction on the channel axis, compressing it to a reduced dimension of C_m, then the Content Encoder performs content encoding, producing a feature map of dimension $\mathrm{H}\times\mathrm{W}\times\sigma^2\times k_{\mathrm{up}}^2$, where σ is the upsampling rate (set

to 2 in this paper) and k_{up} is the upsampling kernel size. The feature tensor undergoes channel-wise unfolding to generate an upsampling kernel of spatial resolution $\sigma H \times \sigma W \times k_{up}^2$, which is subsequently normalized by the Kernel Normalizer. Within the content-adaptive feature reorganization component, the predicted upsampling kernels generated by the kernel estimation unit are spatially correlated with corresponding regions in the input feature space. The region of the original map of size $k_{up} \times k_{up}$ centered on each corresponding point is multiplied element-wise with the predicted upsampling kernel. The final output is a new feature map X' with upsampled dimensions $\sigma H \times \sigma W \times C$.

Compared to nearest neighbor interpolation, CARAFE predicts the upsampling kernels based on the input feature map and performs feature reassembly using these predicted kernels. With a larger receptive field and minimal additional parameters, CARAFE demonstrates superior feature reconstruction quality by intelligently leveraging the inherent semantic content within input feature representations, which are particularly useful for detecting power operation workers. This makes it especially suitable for tasks involving dense small target detection.

2.4 Repulsion-loss loss function

YOLOv8s employs the CIoU (Complete Intersection over Union) loss function to measure IoU between predicted and ground truth bounding boxes. However, CIoU neglects the balance of easy and hard samples, making it more suitable for medium to large targets. For small targets, CIoU is highly sensitive to positional bias, which can degrade performance in power field worker detection tasks involving numerous small and occluded objects.

To tackle this issue, this paper incorporates the Repulsion-Loss [16] function, which consists of three components: an attraction term, a repulsion term, and weighting coefficients. The loss function comprises two components: an attraction component quantifying prediction-GT box alignment, and a repulsion component modeling inter-box exclusion effects. These are balanced through adjustable parameters α and β. Mathematically, the loss function L integrates these components for improved handling of small and occluded targets:

$$L = L_{Attr} + \alpha \cdot L_{RepGT} + \beta \cdot L_{RepBox} \tag{1}$$

Equation (1) formulates the composite loss function with three key components: (1) L_{Attr}, which enforces alignment between the predicted bounding box and its corresponding ground truth; (2) L_{RepGT}, which penalizes excessive overlap between the prediction and neighboring ground truth boxes; and (3) L_{RepBox}, which discourages crowding among predicted detections. The attraction term L_{Attr} is mathematically defined as:

$$L_{Attr} = \frac{\sum_{P \in \rho_+} Smooth_{L1}(B^P, G^P_{Attr})}{|\rho_+|} \tag{2}$$

In equation (2), P denotes a predicted bounding box, ρ_+ represents the set of all positive candidate boxes, B^P is the predicted bounding box after regression, and G^P_{Attr} is the corresponding ground truth box. The $Smooth_{L1}$ distance helps the predicted bounding box to move closer to the true target box.

The mathematical expression for L_{RepGT} is:

$$\begin{cases} L_{\mathrm{RepGT}} = \dfrac{\sum_{\mathrm{p}\in\rho_+} Smooth_{\ln}(IoG(B^{\mathrm{P}}, G^{\mathrm{P}}_{\mathrm{Rep}}))}{|\rho_+|} \\ Smooth_{\ln} = \begin{cases} -\ln(1-x) & x \le \sigma \\ \dfrac{x-\sigma}{1-\sigma} - \ln(1-\sigma) & x > \sigma \end{cases} \\ IoG(B^{\mathrm{P}}, G^{\mathrm{P}}_{\mathrm{Rep}}) = \dfrac{area(B \cap G)}{area(G)} \end{cases} \tag{3}$$

In equation (3), the term $G^{\mathrm{P}}_{\mathrm{Rep}}$ represents the ground truth annotation exhibiting maximal Intersection-over-Ground-Truth (IoU) overlap with the reference bounding box, where IoG quantifies the spatial congruence between predicted box B^{P} and $G^{\mathrm{P}}_{\mathrm{Rep}}$. The Smooth_Ln regularization component imposes a penalty on excessive overlap between detector outputs and adjacent ground truth instances, thereby promoting spatial separation of prediction B^{P} from non-target annotations.

The mathematical expression for L_{RepBox} is:

$$L_{\mathrm{RepBox}} = \frac{\sum_{\mathrm{i}\neq\mathrm{j}} Smooth_{\ln}(IoU(B^{\mathrm{Pi}}, B^{\mathrm{Pj}}))}{\sum_{\mathrm{i}\neq\mathrm{j}} IoU(B^{\mathrm{Pi}}, B^{\mathrm{Pj}}) + a} \tag{4}$$

In equation (4), a is a constant added to avoid division by zero, and B^{Pi} and B^{Pj} are randomly selected predicted bounding boxes of different targets. This repulsion term encourages the overlap area between these two boxes to be as small as possible. To minimize L_{RepBox}, the IoU between the predicted boxes of different targets needs to be minimized.

The L_{RepBox} loss helps reduce the likelihood that predicted boxes with different regression targets are merged into a single detection box after non-maximum suppression (NMS). This results in better robustness of the detector, especially when facing occlusion scenarios.

2.5 DyHead detection head

The YOLOv8s architecture implements a task-decoupled head design, where object detection and classification operations are processed through distinct pathways. In the context of power field operation worker detection, where small targets are abundant, the conventional detection head architecture exhibits constrained multi-scale adaptability and static feature processing characteristics,

potentially compromising detection efficacy. To overcome these limitations, this paper adopts DyHead [17], a dynamic detection module enhanced with an attention mechanism. The DyHead mechanism operates through dynamic feature propagation that fuses scale-sensitive, spatially-aware, and task-oriented attention components, resulting in versatile detection capabilities. Fig. 6 presents its structural organization.

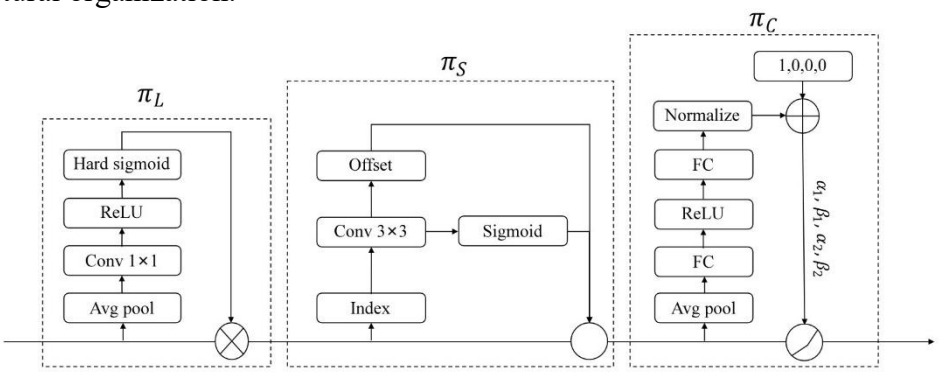

Fig. 6. DyHead structure diagram

Consider a 3D feature tensor $F \in \mathrm{R}^{L\times S\times C}$ extracted from the detection layer, where S denotes spatial dimensions, C represents channel depth, and L indicates the pyramidal level. The DyHead attention mechanism operates on this tensor as specified in Equation (5).

$$W(F) = \pi_{\mathrm{C}}(\pi_{\mathrm{S}}(\pi_{\mathrm{L}}(F)\cdot F)\cdot F)\cdot F \tag{5}$$

In this equation, $\pi_{\mathrm{C}}(\cdot)$, $\pi_{\mathrm{S}}(\cdot)$, and $\pi_{\mathrm{L}}(\cdot)$ correspond to the task-aware attention function, spatial-aware attention function, and scale-aware attention function, respectively. These functions operate on the dimensions C, S, and L, respectively.

Scale-aware attention adjusts the weights of features at different scales within the feature map, enhancing the ability to recognize multi-scale targets. Its calculation process is given in equation (6):

$$\pi_{\mathrm{L}}(F)\cdot F = \sigma(f(\frac{1}{sc}\sum_{S,C}F))\cdot F \tag{6}$$

$$\sigma(x) = \max(0, \min(1, (x+1)/2)) \tag{7}$$

Here, $f(\cdot)$ represents a linear function approximated using a 1×1 convolution, and $\sigma(\cdot)$ denotes the Hard-sigmoid activation function.

Spatial-aware attention improves the model's ability to perceive spatial information in the image, enabling it to more accurately capture the spatial location and shape of the target. The calculation for this attention is shown in equation (8):

$$\pi_S(F)\cdot F=\frac{1}{L}\sum_{l=1}^{L}\sum_{j=1}^{K}w_{l,j}\cdot F(l;p_j+\Delta p_j;c)\cdot\Delta m_j \tag{8}$$

Equation (8) incorporates two key parameters: (1) K, specifying the quantity of sparse sampling points, and (2) $p_j+\Delta p_j$, representing learnable spatial offsets that dynamically attend to salient regions for enhanced feature discrimination. Δm_j represents the learned scalar at location p_j, and both are learned through intermediate layers of the feature map F.

Task-aware attention dynamically adjusts the attention weights for different positions within the feature map, enabling the model to better adapt to detection requirements in complex scenarios. Its calculation process is given in equation (9):

$$\pi_C(F)\cdot F=\max(\alpha_1(F)\cdot F_C+\beta_1(F),\alpha_2(F)\cdot F_C+\beta_2(F)) \tag{9}$$

In equation (9), Let F_C represent the two-dimensional feature matrix extracted from channel C, with $\theta(\cdot)=[\alpha_1,\beta_1,\alpha_2,\beta_2]$ serving as a tunable activation gate function that modulates learning sensitivity. Task-aware attention uses these parameters to apply varying levels of activation to different channels, thereby implementing the attention mechanism.

Compared to the original detection head, DyHead greatly improves the model's representational capability by enabling feature maps to be dynamically adjusted through scale-aware, spatial-aware, and task-aware attention mechanisms. This results in more effective feature representations for detecting workers in power field scenarios, enhancing overall model performance and providing greater accuracy and adaptability in identifying small targets.

3. Experimental results and analysis

3.1 Experimental environment and parameters

The computational experiments were executed on a workstation running Windows 10, utilizing PyTorch 1.9.0 accelerated by CUDA 11.1 for deep learning operations. The system configuration comprised an Intel Xeon Platinum 8350C processor operating at 2.6GHz base frequency paired with an NVIDIA RTX A5000 GPU featuring 24GB of dedicated memory. For model training, we employed stochastic gradient descent optimization with the following parameters: 100 training epochs, mini-batch size of 32 samples, 8 parallel data loader threads, and input image dimensions standardized to 640×640 pixels through resizing. The optimizer was configured with an initial learning rate of 0.01, momentum coefficient of 0.937, and L2 regularization weight decay of 0.0005 to prevent overfitting during the training process.

3.2 Experimental dataset

The dataset for this study was compiled through manual online searches and collaboration with power companies. Online searches were conducted using a predefined set of keywords (e.g., "power utility worker," "helmet wearing electrician") to gather diverse images. The raw collection was then filtered by three annotators based on criteria of image clarity and the presence of clearly identifiable personnel. The final dataset consists of 1,200 images. To improve generalization, data augmentation strategies including random rotations, scaling, and color jittering were employed during training. The dataset was randomly split into training, validation, and test sets with a 6:2:2 ratio (720/240/240 images).

3.3 Experimental evaluation metrics

The detection framework's performance is quantitatively assessed through five key indicators: precision (P), recall (R), average precision (AP), parameter count (Params), and computational complexity (GFLOPs). Mathematical formulations for P, R, and AP appear in Equations (10)-(13).

$$P = \frac{N_{\mathrm{TP}}}{N_{\mathrm{TP}} + N_{\mathrm{FP}}} \tag{10}$$

$$R = \frac{N_{\mathrm{TP}}}{N_{\mathrm{TP}} + N_{\mathrm{FN}}} \tag{11}$$

$$P_{\mathrm{mAP}} = \frac{\sum_{i=1}^{k} P_{A_i}}{k} \tag{12}$$

$$P_{\mathrm{A}} = \int_0^1 P(t)dt \tag{13}$$

In the formula, Let N_{TP} be the true positive count, N_{FP} the false positive instances, and N_{FP} the false negative cases. For multi-class evaluation, P_{A} denotes class-specific average precision, where $\mathrm{i} \in \{1,...,\mathrm{k}\}$ represents k distinct classes. The mean average precision P_{mAP} is computed as $(1/\mathrm{k})\sum \mathrm{P}_{\mathrm{A_i}}$ across all classes.

3.4 Ablation study

To assess the impact of our architectural enhancements on detection capability, we conducted systematic component-wise analyses using YOLOv8s as the baseline framework. The quantitative findings from these controlled experiments are presented in Table 1.

Table 1

The ablation experiment results of YOLOv8s

Groups	Method	P/%	R/%	mAP/%	Params/10^6	GFLOPs
1	YOLOv8s	85.22	81.25	85.42	**11.12**	**28.7**
2	+SGAM_C2f	87.25	83.51	87.53	13.05	31.3
3	+SGAM_C2f+CARAFE	88.11	84.05	88.31	13.25	31.6
4	+SGAM_C2f+CARAFE+Repulsion-Loss	88.73	85.59	89.51	13.25	31.6
5	+SGAM_C2f+CARAFE+Repulsion-Loss+DyHead	**89.56**	**86.47**	**90.23**	15.05	32.9

Through analysis of Table 1, we can draw the following conclusions: The first experiment utilized the original YOLOv8s algorithm, achieving an accuracy of 85.22%, a recall rate of 81.25%, a mean average precision (mAP) of 85.42%, with a parameter count of 11.12M and a computational cost of 28.7GFLOPs. In the second experiment, the GAM attention mechanism was introduced and the C2f module was redesigned using SPDConv. The improved SGAM_C2f module led to an increase in accuracy, recall rate, and mAP by 2.03%, 2.26%, and 2.11% respectively, while the parameter count and computational cost increased by 1.93M and 2.6GFLOPs respectively. The third experiment further introduced the lightweight upsampling operator CARAFE to replace the original upsampling method, resulting in an improvement in accuracy, recall rate, and mAP by 0.86%, 0.54%, and 0.78% respectively, with an additional increase in parameter count and computational cost of 0.20M and 0.3GFLOPs respectively. In the fourth experiment, based on the third experiment, the Repulsion-Loss function was employed, leading to an improvement in all metrics, with accuracy, recall rate, and mAP increasing by 0.62%, 1.54%, and 1.20% respectively, while the parameter count and computational cost remained unchanged. Finally, in the fifth experiment, the DyHead detection head was used to replace the original detection head, resulting in an increase in accuracy, recall rate, and mAP by 0.83%, 0.88%, and 0.72% respectively, with an additional increase in parameter count and computational cost of 1.8M and 1.3GFLOPs respectively.

Overall, the SCR-YOLOv8s algorithm achieved gains of 4.34%, 5.22%, and 4.81% in precision, recall, and mAP, respectively, over the baseline. The associated increases in parameter count and computational cost (3.93M parameters and 4.2 GFLOPs) are justified for our target application in power-field scenarios. This trade-off is strategically made because the significant enhancement in detection reliability for safety-critical monitoring outweighs the computational costs, especially since the model is designed for deployment on edge computing devices where a high frame rate is not imperative and sub-real-time processing is fully sufficient.

To visually demonstrate the difference in detection performance between the YOLOv8s algorithm and the SCR-YOLOv8s algorithm, the original model and

the improved model were tested using the test set from the electric power field operator detection dataset. The test results are shown in Fig. 7.

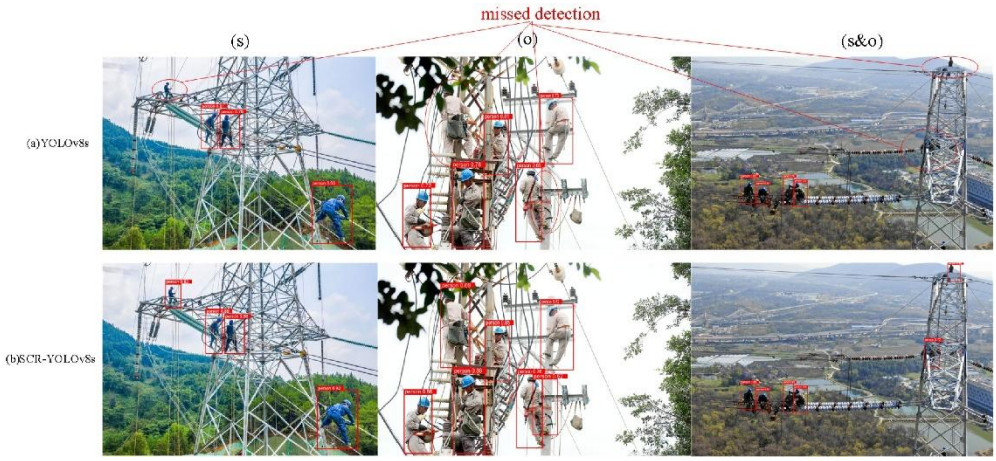

Fig. 7. Detection effect comparison of different models

Among the test data, the first group (s) consists of small targets, the second group (o) consists of occluded targets, and the third group (s&o) includes both small and occluded targets. Analysis of Fig. 7 reveals that compared to Fig. 7(a), Fig. 7(b) detects more small and occluded targets. For the first experimental group (s), when the target size is particularly small, the SCR-YOLOv8s algorithm still accurately detects the power field operator in the top left corner. For the second experimental group (o), when targets are mutually occluded, the SCR-YOLOv8s algorithm accurately detects the other two occluded targets. For the third experimental group (s&o), in scenarios where the target size is very small and occlusion exists, the SCR-YOLOv8s algorithm significantly reduces missed detections. In general, the SCR-YOLOv8s algorithm exhibits significant improvements in the detection of both small and occluded targets.

3.5 Comparative experiments

To validate the enhanced detection capability of our optimized YOLOv8s architecture for power industry personnel identification, we performed comprehensive benchmarking against state-of-the-art detection frameworks. The evaluation compared seven representative models: Faster R-CNN, YOLOv3-tiny, YOLOv5s, YOLOv7-tiny, YOLOv7, baseline YOLOv8s, and our proposed SCR-YOLOv8s. Performance assessment employed five quantitative measures: detection accuracy (P), coverage rate (R), comprehensive precision metric (mAP), model footprint (Params), and computational demand (GFLOPs), with detailed comparative results documented in Table 2.

Table 2

Performance comparison of different models

Model	P/%	R/%	mAP/%	Params/10^6	GFLOPs
Faster R-CNN	82.23	74.12	79.65	40.45	371.0
YOLOv3-tiny	79.92	72.21	78.66	8.69	**13.0**
YOLOv5s	84.85	80.75	84.87	7.08	16.6
YOLOv7-tiny	82.32	79.63	83.32	**6.03**	13.3
YOLOv7	85.62	82.65	86.77	37.24	105.3
YOLOv8s	85.22	81.25	85.42	11.12	28.7
SCR-YOLOv8s	**89.56**	**86.47**	**90.23**	15.05	32.9

From the analysis of Table 2, the following observations can be made: Due to the large number of small targets in the dataset, Faster R-CNN and YOLOv3-tiny exhibit relatively weak detection performance. Although YOLOv7 achieves favorable results, its high model complexity and computational demand make it less suitable for applications requiring lightweight and real-time processing in power field operator detection tasks. The detection accuracy of YOLOv7-tiny and YOLOv5s is slightly lower than that of YOLOv8s. YOLOv8s delivers a good balance between detection performance and model efficiency, making it a suitable baseline algorithm for this specific scenario. The improved SCR-YOLOv8s model achieves the best overall performance in terms of precision, recall, and mean average precision. Compared to the baseline, its parameter count and computational cost increase by 3.93M and 4.2 GFLOPs, respectively, yet it still maintains a relatively compact and efficient structure. Among all tested models, SCR-YOLOv8s demonstrates the most outstanding detection capability, significantly enhancing performance while preserving real-time execution.

4. Conclusion

To boost detection performance for electric power field operators, the SCR-YOLOv8s algorithm is proposed. It incorporates the SGAM_C2f structure in the backbone, combining GAM attention and SPDConv to enhance small target feature extraction. The model also uses the lightweight CARAFE upsampling operator to extend its receptive field. To improve bounding box comparison fairness and handle occlusion problems, the Repulsion-Loss function is introduced. Furthermore, DyHead upgrades the original detection head, significantly boosting small target detection capabilities. Experimental results indicate that SCR-YOLOv8s achieves notable improvements in precision, recall, and mAP by 4.34%, 5.22%, and 4.81% respectively compared to YOLOv8s, while preserving its lightweight and real-time performance. This enhancement effectively improves detection of small and blurred targets, which is critical for complex power site environments. Future work will aim to further refine the model's ability to detect blurred targets from camera-captured images.

Acknowledgments

This research was funded by Guizhou Power Grid Co. Ltd, grant number 060000KC23100011.

REFERENCES

[1] *Girshick R*. Fast r-cnn//Proceedings of the IEEE international conference on computer vision. 2015: 1440-1448.

[2] *Ren S, He K, Girshick R*, et al. Faster r-cnn: Towards real-time object detection with region proposal networks. lEEE Transactions on Pattem Analysis and Machine Intelligence, 2017,39(6): 1137-1149.

[3] *Liu W, Anguelov D, Erhan D*, et al. Ssd: Single shot multibox detector//Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14. Springer International Publishing, 2016: 21-37.

[4] *Redmon J, Divvala S, Girshick R*, et al. You only look once: Unified, real-time object detection//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 779-788.

[5] *Jia H J,Huang X H,Xiong M*, et al. Research on lightweight safety helmet detection algorithm based on knowledge distillation. Power Systems and Big Data, 2025 ,28(10):36-43.

[6] *Liu M, Hu X, Wan X*, et al. Research on the detection algorithm of electric workers in the limited spaces of hydroelectric power station. Energy Reports, 2024, 12: 472-480.

[7] *Li Y, Hu Z, Zhang Y, Liu J, Tu W, Yu H.* DDEYOLOv9: Network for Detecting and Counting Abnormal Fish Behaviors in Complex Water Environments. Fishes. 2024; 9(6):242. https://doi.org/10.3390/fishes9060242

[8] *Wang C Y, Liao H Y M, Wu Y H*, et al. CSPNet: A new backbone that can enhance learning capability of CNN//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops. 2020: 390-391.

[9] *Wang C Y, Bochkovskiy A, Liao H Y M*. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2023: 7464-7475.

[10] *Wang W, Xie E, Song X*, et al. Efficient and accurate arbitrary-shaped text detection with pixel aggregation network//Proceedings of the IEEE/CVF international conference on computer vision. 2019: 8440-8449.

[11] *Li X, Wang W, Wu L*, et al. Generalized focal loss: Learning qualified and distributed bounding boxes for dense object detection. Advances in Neural Information Processing Systems, 2020, 33: 21002-21012.

[12] *Zheng Z, Wang P, Liu W*, et al. Distance-IoU loss: Faster and better learning for bounding box regression//Proceedings of the AAAI conference on artificial intelligence. 2020, 34(07): 12993-13000.

[13] *Kirkland E J, Kirkland E J*. Bilinear interpolation. Advanced computing in electron microscopy, 2010: 261-263.

[14] *Carlson R E, Fritsch F N*. Monotone piecewise bicubic interpolation. SIAM journal on numerical analysis, 1985, 22(2): 386-400.

[15] *Wang J, Chen K, Xu R*, et al. Carafe: Content-aware reassembly of features//Proceedings of the IEEE/CVF international conference on computer vision. 2019: 3007-3016.

[16] *Wang X, Xiao T, Jiang Y*, et al. Repulsion loss: Detecting pedestrians in a crowd//Proceedings of the IEEE conference on computer vision and pattern recognition. 2018: 7774-7783.

[17] *He K, Zhang X, Ren S*, et al. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification//Proceedings of the IEEE international conference on computer vision. 2015: 1026-1034.