Cementite Detection in Spherical Pearlite Based on YOLOv9

Lichuan Liu, Yujun Zhang*, Xusheng Li, Yixiao Sun, Zhigao Zhao, Chengshuang Yu, Lingchu Wang, Dongying Ju

Abstract—**In modern materials science, metallographic analysis is an important method for evaluating the microstructure and properties of materials. After spheroidizing annealing, the morphology and distribution of pearlite in steel have a significant impact on the subsequent heat treatment of the material. This paper constructs a dataset of cementite in spherical pearlite and proposes a YOLOv9-based method for detecting cementite. This method innovatively introduces the fusion-enhanced module C3_CD_CBAM to enhance the model's feature extraction capability. Built upon the C3 structure, this module sequentially integrates channel attention and spatial attention, adaptively optimizing feature weight distribution and improving detection accuracy. Additionally, the introduction of the CARAFE module further enhances the spatial resolution of feature maps, significantly improving performance across various visual tasks. The network also incorporates the SimAM mechanism, which adaptively adjusts attention weights to enhance feature representation, improving detection accuracy without introducing extra parameters. Through experiments, we demonstrate the effectiveness of the improved YOLOv9 model, achieving an average accuracy of 89.9% on the proposed dataset—an improvement of 1.2% over the baseline YOLOv9 model. Leveraging the latest advancements in deep learning architectures and data augmentation techniques, this study enhances the automation and accuracy of spherical pearlite evaluation, providing a novel solution for metallographic analysis.**

Index Terms—**Steel, Metallography, Spherical cementite, Detection, YOLOv9.**

I. Introduction

STEEL is an essential raw material in mechanical design and manufacturing. After spheroidizing annealing, the layered or networked carbides within steel aggregate into spherical shapes, improving the material's machinability by reducing hardness and refining the microstructure[1–3]. This structure, known as spheroidal pearlite, consists of a ferrite matrix and spherical cementite[4]. To comprehensively evaluate the properties of steel based on the morphology and distribution of spheroidal pearlite, researchers systematically classify the quantity and distribution of cementite within spheroidal pearlite. Traditional classification methods for detecting spheroidal cementite typically rely on manual observation, which is not only inefficient but also prone to errors due to visual fatigue under prolonged, high-intensity work [5]. With the rapid development of deep learning technology, computer vision-based methods have gradually become a new trend in metallographic structure detection and analysis [6, 7].

The development of deep learning-based object detection techniques represents a paradigm shift from handcrafted feature engineering to data-driven methodologies. The core breakthroughs lie in the collaborative optimization of feature representation and computational architecture[8]. Traditional methods rely on manually designed feature descriptors, such as Haar cascades, Histogram of Oriented Gradients (HOG), and Support Vector Machine (SVM) classifiers, which locate objects using a sliding window approach. However, these methods are constrained by limited feature generalization capabilities and excessive computational redundancy. The introduction of Convolutional Neural Networks (CNNs) has redefined feature extraction mechanisms. Hierarchical convolution and pooling operations, through local receptive fields, enable progressive abstraction from edge textures to semantic structures, laying the foundation for multi-scale object detection in complex scenarios [9, 10]. Two-stage detection frameworks, exemplified by the R-CNN series, generate candidate regions using a Region Proposal Network (RPN) followed by refined classification and regression, offering significant accuracy advantages due to their cascade-based design. Fast R-CNN enhances efficiency by incorporating Region of Interest (ROI) Pooling for feature sharing, while Faster R-CNN further integrates candidate box generation into an end-to-end training process, significantly improving detection speed [11]. In contrast, single-stage detection models, represented by the YOLO series and SSD, adopt a global regression strategy. By combining grid-based spatial partitioning and multi-scale feature pyramids (FPN), these models unify bounding box prediction and class determination within a single forward pass, achieving real-time inference while maintaining high mean Average Precision (mAP) [12].

The continued evolution of the YOLO series focuses on optimizing the trade-off between accuracy and speed. Key

Manuscript received April 1, 2025; revised June 10, 2025.

This work was supported by the Key Laboratory of Internet of Things Application Technology on Intelligent Construction, Liaoning Province (2021JH13/10200051)

Lichuan Liu is a graduate student of School of Computer and Software Engineering, University of Science and Technology Liaoning, Anshan 114051, China (e-mail: 1505180917@qq.com).

Yujun Zhang is a Professor of School of Computer Science and Software Engineering, University of Science and Technology Liaoning, Anshan 114051, China (Corresponding Author, e-mail: 1997zyj@163.com).

Xusheng Li is the Chief Engineer of Intelligent Heat Treatment Manufacturing at Zhejiang XCC Group CO., Ltd. Xinchang 312500, China (e-mail: lixushengt@sina.com).

Yixiao Sun is a Ph.D. candidate of School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan 114051, China (e-mail: robinsunyixiao@163.com).

Zhigao Zhao is the Manager of the Technology Department, Heat Treatment Branch at Zhejiang XCC Group CO., Ltd. Xinchang 312500, China (e-mail: zhaozhigao@xcc-zxz.com).

Chengshuang Yu is the Process Engineer at Zhejiang XCC Group CO., Ltd. Xinchang 312500, China (e-mail: yuchengshuang@xcc-zxz.com).

Lingchu Wang is the Director of the Testing Center at Zhejiang XCC Group CO., Ltd. Xinchang 312500, China (e-mail: wanglingchu@xcc-zxz.com).

Dongying Ju is a Professor at the University of Science and Technology Liaoning, Anshan 114051, China (e-mail: dyju.sitec@gmail.com).

advancements include the lightweight Cross-Stage Partial Network (CSPNet), adaptive anchor box clustering algorithms, and dynamic label assignment strategies. Furthermore, by integrating Visual Language Models (VLMs) for cross-modal feature fusion, the detection accuracy of occluded objects and small-scale instances has been significantly improved. As a highly efficient object detection model, YOLOv9 exhibits superior speed and precision, making it particularly suitable for real-time image analysis. Therefore, this paper proposes a YOLOv9-based method for detecting cementite in spheroidal pearlite. The improved YOLOv9 model demonstrates enhanced performance in multi-scale and complex environment object detection tasks, significantly boosting overall detection accuracy.This method enables accurate and rapid detection of the quantity and distribution of spheroidal cementite in samples, thereby assisting inspectors in completing grading tasks with greater efficiency and precision.

The improved YOLOv9 model proposed in this paper introduces the C3_CD_CBAM fusion module into the neck network. This module optimizes multimodal feature representation through a multi-dimensional attention coordination mechanism and cross-scale feature interaction strategy. Based on the classic Cross Stage Partial (CSP) architecture, it incorporates a dual-path attention-guided residual learning paradigm for both channel and spatial dimensions, constructing a composite feature fusion unit with dynamic feature selection and contextual awareness capabilities. By deeply integrating the Convolutional Block Attention Module (CBAM) with cross-stage downsampling (CD) operations, the model forms a multi-level feature enhancement system.

In the neck network, the upsampling module is replaced with the CARAFE module, which significantly improves the spatial resolution of feature maps through adaptive interpolation and reassembly techniques. The SimAM (Simple Attention Module) is introduced after RepNCSPELAN4 to optimize feature representation and improve detection accuracy. SimAM adaptively allocates attention weights by calculating the variance distribution within the feature map, without requiring additional trainable parameters, thus enhancing the response of target regions and suppressing irrelevant background. It has lower computational overhead, and works in synergy with RepNCSPELAN4 and CBFuse to optimize feature fusion strategies, improving the detection capability of small targets.

II. Related work

Metallography is the study of the structure of metals and alloys. Metallographic analysis can be regarded as a detection tool to assist in identifying a metal or alloy, to evaluate whether an alloy is processed correctly, to inspect multiple phases within a material, to locate and characterize imperfections such as voids or impurities, or to find the damaged areas of metallographic images [13, 14]. The preparation of metallographic samples is a prerequisite for conducting metallographic analysis. It is crucial to select and prepare representative samples. Typically, the preparation of metallographic samples involves the following steps: sampling, embedding (which can sometimes be omitted), grinding (coarse grinding and fine grinding), polishing, and etching.

The detection work is generally carried out by manually observing the surface of metallographic specimens under a microscope to qualitatively describe the microstructural features of the metal material, or by comparing with various standard images to assess the microstructure. This method often involves subjectivity and lacks high reproducibility and accuracy.

With the development of computer vision technology, deep learning-based object detection methods have shown great potential in the field of metallographic analysis, among which the YOLO series has been widely applied due to its efficient detection capability and excellent real-time performance. Against this backdrop, YOLOv9 has further improved detection accuracy and inference speed through multiple structural optimizations while inheriting the efficient detection framework of the YOLO architecture [15]. Its main architectural features include the introduction of Dynamic Reparameterization, which allows multi-branch structures during training to be merged into a single path during inference, thereby reducing computational overhead and improving inference efficiency. Additionally, it adopts an improved Neck structure, such as a more efficient feature fusion module, to enhance multi-scale object detection capabilities, and incorporates a lightweight attention mechanism in the Backbone to improve feature representation. In the Head, the Anchor mechanism has been optimized to make bounding box regression more precise and generalizable. Moreover, YOLOv9 introduces advancements in training strategies, such as a new loss function that optimizes bounding box matching, improving the detection accuracy of small and overlapping objects. Compared to previous YOLO models, YOLOv9 has shown significant improvements in mean Average Precision (mAP) on datasets such as COCO and VOC while maintaining superior real-time performance in terms of computational complexity. Compared to YOLOv8, its enhanced feature extraction structure allows for higher detection accuracy with the same computational resources, while relative to YOLOv7, its optimized Neck structure enhances the perception of small objects, making it more advantageous in applications such as remote sensing imagery, autonomous driving, and industrial inspection. Furthermore, compared to YOLOv5, YOLOv9 offers faster inference speed and better object recognition performance in complex backgrounds [16–19]. These advantages enable YOLOv9 to maintain a lightweight architecture while excelling in high-precision object detection tasks. Therefore, we have chosen YOLOv9 as the baseline model and will further optimize it to better suit our specific detection objectives.

III. Method Introduction

A. Modules of the Improved YOLOv9 Algorithm

While YOLOv9 can enhance detection accuracy by optimizing the object localization and classification loss functions, its performance may degrade when handling dense and small targets. This is because the model struggles to precisely identify and distinguish closely packed cementite particles, especially when they are connected. Additionally, YOLOv9 has limited ability to separate adjacent targets due to the insufficient resolution of feature maps, making it difficult to effectively detect individual cementite particles

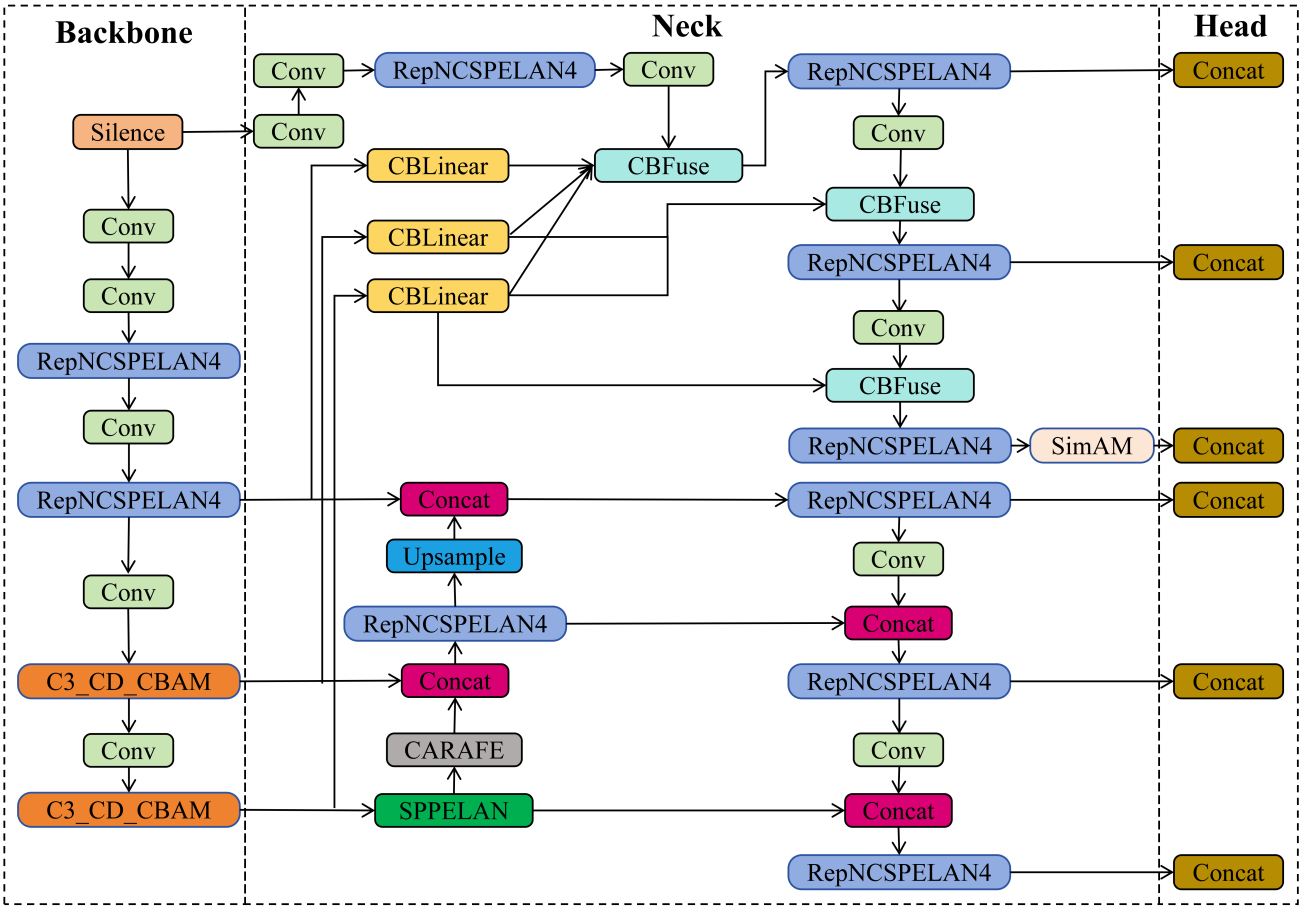

Fig. 1: Overall improved architecture diagram

in high-density regions. Furthermore, the model may fail to capture subtle structural differences within cementite formations, leading to inaccurate detections. Therefore, although YOLOv9 demonstrates strong detection capabilities in many scenarios, further improvements and optimizations are necessary to effectively handle the challenges posed by the cementite dataset.

To address these challenges, this paper proposes an enhanced structure for YOLOv9. The improved YOLOv9 architecture is shown in Figure 1. First, the RepNCSPELAN4 module in the backbone network is replaced with the C3_CD_CBAM module, where C3_CD enhances multi-scale feature fusion, and the CBAM attention mechanism optimizes key region perception, improving fine-grained target recognition. Second, the traditional upsampling module is replaced with CARAFE, which enhances spatial resolution through content-aware feature reassembly, reducing small-target information loss and improving localization accuracy. Finally, the SIMAM module is integrated into the detection head to enhance feature discrimination, suppress background interference, and improve detection robustness. These improvements enhance YOLOv9's accuracy and adaptability for cementite detection.

B. C3_CD_CBAM

The C3_CD_CBAM module first inherits the structure of the C3 (Cross-Stage Partial) module. The C3 module efficiently performs feature fusion by partitioning the input feature map into groups, avoiding potential gradient vanishing problems in traditional convolutional networks. The C3 module divides the feature map into two parts, each undergoing a convolution operation, followed by concatenation of the results. The module further optimizes feature extraction efficiency through residual connections and bottleneck structures. The architecture of the C3_CD_CBAM module is shown in Figure 2. The core feature fusion process in C3 can be expressed as:

$$Y = \text{Conv}_3\left(\text{concat}\left(\mathcal{F}_B\left(\text{Conv}_1(X)\right), \text{Conv}_2(X)\right)\right) \quad (1)$$

where X is the input feature map, $Conv_1$, $Conv_2$ and $Conv_3$ are convolutional operations, $\mathcal{F}_B$ represents the Bottleneck transformation containing multiple 1×1 and 3×3 convolutions, and concat$(\cdot)$ denotes channel-wise concatenation. Equation (1) describes how the C3 module efficiently integrates information from two feature branches, enhancing feature representation.

The CBAM module enhances the model's attention to critical feature regions. CBAM (Convolutional Block Attention Module) introduces both channel and spatial attention mechanisms, enabling the model to focus more on important areas and channels in the image while suppressing background noise and irrelevant features, thereby improving the detection performance [20]. Specifically, the CBAM module first extracts global maximum and average information from the feature map through adaptive max-pooling and average-pooling operations. These features are then fused via fully

connected layers (FC) to generate the channel attention map. The channel attention mechanism is computed as follows:

$$\alpha_c = \sigma\left(W_2 \cdot \mathrm{ReLU}(W_1 \cdot (M_{\text{avg}} + M_{\text{max}}))\right) \tag{2}$$

where M_{avg} and M_{max} are the features obtained through average-pooling and max-pooling, W_1 and W_2 are trainable parameters, and σ is the Sigmoid activation function. Equation (2) shows that the channel attention map α_c is generated by applying a ReLU activation to the sum of the average and max-pooled features, followed by two fully connected layers. This attention mechanism allows the network to focus on the most informative channels.

The spatial attention mechanism is computed as follows:

$$\alpha_s = \sigma\left(\mathrm{Conv}(\mathrm{concat}(M_{\text{avg}}, M_{\text{max}}))\right) \tag{3}$$

where M_{avg} and M_{max} are the spatial features obtained through global average-pooling and max-pooling, respectively. The concatenated features are processed by a convolution operation to generate the spatial attention map α_s. As shown in equation (3), the spatial attention map α_s helps the model focus on relevant spatial regions, enhancing object localization.

The overall structure of the C3_CD_CBAM module combines the C3 module and the CBAM module, fully utilizing the advantages of both. It preserves the feature fusion efficiency of the C3 module while enhancing the feature map's attention mechanism through CBAM. This significantly improves the model's detection accuracy in complex scenarios. The module is particularly suitable for small object detection tasks, such as cementite detection, and enhances the model's ability to perceive small-sized objects and robustness.

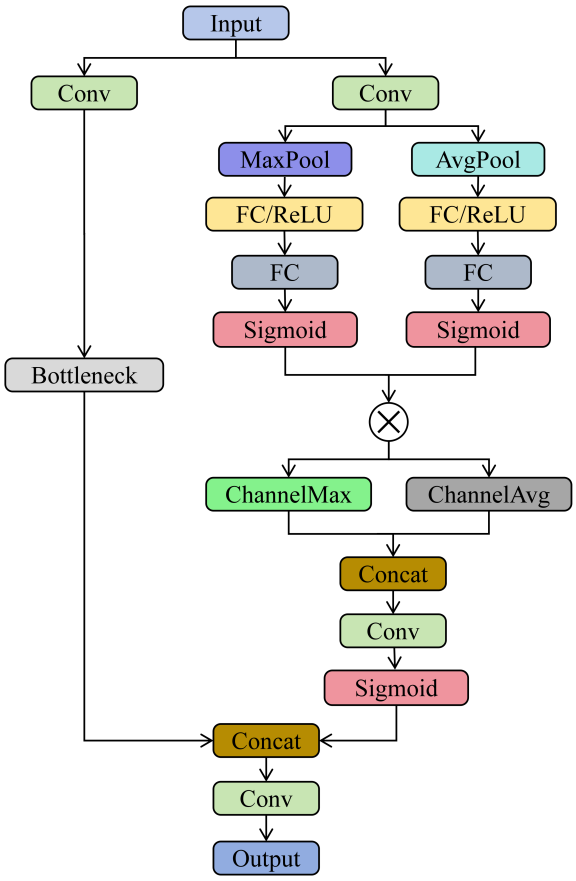

Fig. 2: C3_CD_CBAM network architecture

C. CARAFE

The CARAFE (Content-Aware ReAssembly of FEatures) module is an advanced upsampling method that enhances the preservation of fine details in feature maps. Unlike traditional upsampling approaches such as bilinear interpolation or transposed convolution, CARAFE generates adaptive reassembly weights dynamically based on the input feature content, thereby improving feature recovery. The module consists of four key stages: channel compression, weight encoding, feature unfolding, and feature reassembly [21, 22]. Its architecture is shown in Figure 3.

The first step is channel compression, where a 1×1 convolution is applied to the input feature map to reduce its channel dimension. The channel compression operation can be expressed as follows:

$$W = \mathrm{Conv}_{1\times1}(X) \tag{4}$$

where X is the input feature map, and W is the intermediate feature map obtained by applying a 1×1 convolution to reduce the number of channels. As shown in equation (4), this step helps to decrease computational complexity while retaining key feature information.

The second step is weight encoding, where a convolution operation is applied to the intermediate feature map. The output is then processed using a PixelShuffle operation, producing a preliminary reassembly weight map. This weight map is further normalized using the Softmax function to generate the final reassembly weight map:

$$K' = \sigma\left(\mathrm{Conv}_{k_{\text{enc}}}(W)\right) \tag{5}$$

where σ represents the Softmax activation function, and $\mathrm{Conv}_{k_{\text{enc}}}$ denotes a convolution operation with kernel size k_{enc}. As shown in equation (5), the reassembly weight map is generated by applying a convolutional layer followed by a softmax operation, ensuring that the learned weights are spatially adaptive.

The third step is feature unfolding, where the input feature map is upsampled using nearest-neighbor interpolation and then unfolded using a sliding window operation. This extracts local region features, which can be expressed as follows:

$$X' = \mathrm{Unfold}(\mathrm{Upsample}(X)) \tag{6}$$

where X is the input feature map, and X' represents the unfolded features obtained after upsampling and applying a sliding window operation. As shown in equation (6), this step enables the extraction of local neighborhood features, which are later reassembled using adaptive weights.

The final step is feature reassembly, where the unfolded features are weighted by the reassembly weights. The final output feature map is obtained through an element-wise weighted sum:

$$Y = \sum_{i=1}^{k_{\text{up}}^2} K_i' \cdot X_i' \tag{7}$$

where K_i' and X_i' represent individual elements of the weight map and unfolded feature map, respectively. As shown in

equation (7), the final feature map Y is obtained by adaptively reassembling local features using learned attention-based weights.

The overall structure of the CARAFE module enhances the upsampling process by incorporating content-aware adaptive weight generation. It effectively preserves fine details and enhances feature map resolution, making it particularly useful for small object detection tasks, such as cementite detection in metallographic images. By dynamically reassembling features based on content, CARAFE improves the model's ability to recover spatial details, ultimately enhancing detection robustness in complex environments.

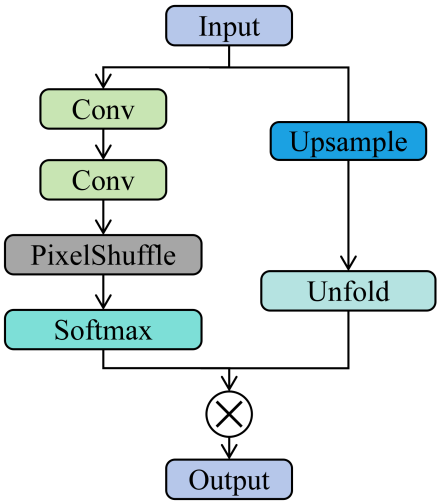

Fig. 3: CARAFE network architecture

D. SimAM

SimAM (Similarity-based Attention Module) is a lightweight and efficient attention mechanism that enhances feature representation by refining spatial and channel-wise information. Unlike traditional attention mechanisms, SimAM does not require additional learnable parameters or complex operations such as convolutions or pooling layers. Instead, it applies a simple yet effective activation function to adjust the feature responses.

The core idea of SimAM is to compute the variance of each feature map across spatial dimensions and use this variance to generate an attention map. The module first calculates the squared difference between the feature map and its spatial mean:

$$X' = (X - \mu_X)^2 \tag{8}$$

where X is the input feature map, and μ_X represents the mean value of X across the spatial dimensions.

Next, SimAM normalizes the variance using the total sum of squared differences and a small constant λ to avoid numerical instability. The normalization process is expressed as follows:

$$S = \frac{X'}{4\left(\frac{\sum X'}{N} + \lambda\right)} + 0.5 \tag{9}$$

where N is the total number of spatial positions minus one, and λ is a small regularization term.

Finally, the attention map is generated using a Sigmoid activation function, and the input feature map is scaled accordingly:

$$Y = X \cdot \sigma(S) \tag{10}$$

where $\sigma(S)$ represents the Sigmoid activation applied to the normalized variance, and Y is the final output feature map.

SimAM efficiently enhances the feature representation by leveraging spatial variance information without introducing additional computational overhead. By applying element-wise modulation, the module refines feature maps while maintaining simplicity and efficiency.

IV. Experimental Design and Implementation

A. Dataset Introduction

The metallographic images containing spherical cementite used in this paper come from the Metallographic Laboratory of Zhejiang XCC Group CO., Ltd. After applying Gaussian denoising and cropping, 120 images with a size of 640px * 640px were created. The spherical cementite in the images was annotated using the T-rex label tool, the annotation example of this dataset is shown in Figure 4. The labeled COCO format files were then converted into the label format required for YOLOv9 training. After completing the above processing, the dataset was divided into a training set and a validation set, with a ratio of 8 : 2, containing one category: spherical cementite.

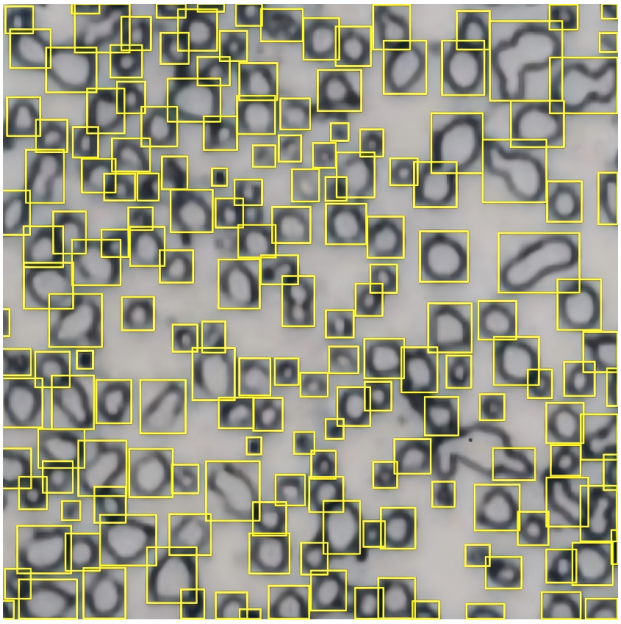

Fig. 4: Annotation examples in our dataset.

B. Experimental environment and parameter configuration

The experiments in this study were conducted on a server equipped with an NVIDIA GeForce RTX 3080Ti graphics card, which has 10GB of VRAM, effectively supporting the efficient training of deep learning models. The operating system was Windows 11, and the main software environment included CUDA 11.8, Python 3.8.10, and Pytorch 2.0.0. The model training was set for a total of 300 epochs. The batch size was set to 4. Additionally, the learning rate was set to

0.01 to balance training speed and model convergence. Other parameters were kept at their default values.

C. Model evaluation metrics

Single-class detection, as a specific binary classification problem, evaluates the model's performance on this class using four elements: True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN).

In this experiment, Precision (P) and Recall (R) were chosen as the evaluation metrics for spherical cementite detection results. Precision indicates the proportion of positive samples among the predicted positive samples, as shown in formula (11).

$$P = \frac{TP}{TP + FP} \tag{11}$$

Recall indicates the ratio of correctly predicted positive samples to the total labeled positive samples, as shown in formula (12).

$$R = \frac{TP}{TP + FN} \tag{12}$$

Precision (P) and Recall (R) are negatively correlated. To comprehensively assess the quality of the algorithm, the Precision-Recall (PR) curve is typically plotted with Recall on the x-axis and Precision on the y-axis.

YOLOv9 baseline model and the improved model's PR curve are shown in Figure 5. The red curve represents the baseline model, while the blue curve represents the improved model. The Precision-Recall (P-R) curve in the figure illustrates the differences in detection performance of the YOLOv9 model on the cementite dataset before and after improvements. The improved YOLOv9 model shows an increase in the mean Average Precision (mAP@0.5) for this category from 0.887 to 0.899, with significant enhancements in both detection accuracy and recall.

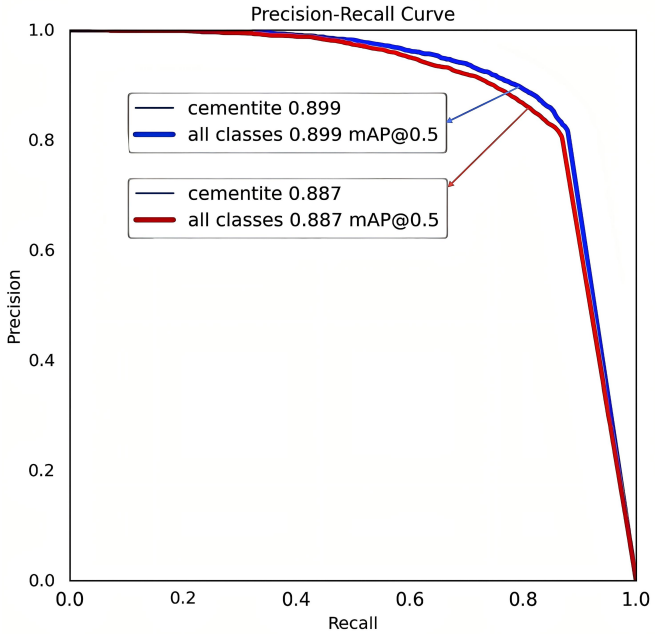

Fig. 5: Precision-Recall Curve

D. Comparison results of different models

To comprehensively evaluate the performance of our proposed YOLOv9-based object detection model on metallographic images, we conducted detailed experiments on the cementite dataset. Table 1 summarizes the experimental results of various models, systematically comparing Image Size, Precision, Recall, and mean Average Precision (mAP). The results show that our model performs excellently across all metrics, particularly achieving an mAP of 89.9%, surpassing the baseline YOLOv9 model (88.7%) as well as other models such as YOLOv8, YOLOv10, YOLOv11 and yolov12 . Overall, our improved model demonstrates significant accuracy improvements on the cementite dataset, laying a solid foundation for future research and applications.

TABLE I: Compare Different categories pairwise

Method	ImageSize	Precision	Recall	mAP
YOLOv8	640*640	88.2	88.0	87.9
YOLOv9	640*640	91.4	87.0	88.7
YOLOv10	640*640	94.2	86.0	88.1
YOLOv11	640*640	85.9	86.0	88.5
YOLOv12	640*640	90.9	86.0	88.9
ours	640*640	92.6	88.0	89.9

E. Ablation experiments

To verify the impact of each module on the performance of our proposed improved YOLOv9 model on the cementite dataset, we conducted ablation experiments, with the results shown in Table 3. In this ablation study, we separately tested the C3_CD_CBAM module, SARAFE module, and SimAM module. By progressively adding or removing these modules, we were able to observe their influence on the overall performance of the model. The experimental results in the table display the Precision, Recall, and mean Average Precision (mAP) for different combinations of modules.

The baseline model achieves performance of 91.4%, 87.0% and 88.7% in terms of Precision, Recall and mAP, respectively. Experimental results indicate that introducing the C3_CD_CBAM module leads to a slight decrease in Precision by 1.9 percentage points but effectively improves mAP to 89.0%. Meanwhile, the CARAFE feature re-sampling technique enhances mAP by 0.6 percentage points to 89.3% while maintaining Recall stability. Notably, the SimAM attention mechanism, when applied independently, causes a significant drop in Precision by 8.7 percentage points but still preserves the robustness of the mAP metric.

Further analysis reveals that the combined application of CARAFE and SimAM exhibits a significant synergistic effect, boosting Precision to a breakthrough level of 91.9% and increasing mAP by 1.0 percentage point to 89.7%. When fully integrating C3_CD_CBAM, CARAFE and SimAM, the model achieves optimal overall performance: Precision increases by 1.2 percentage points to 92.6%, Recall remains stable at 88.0%, and mAP improves by 1.2 percentage points to 89.9%.

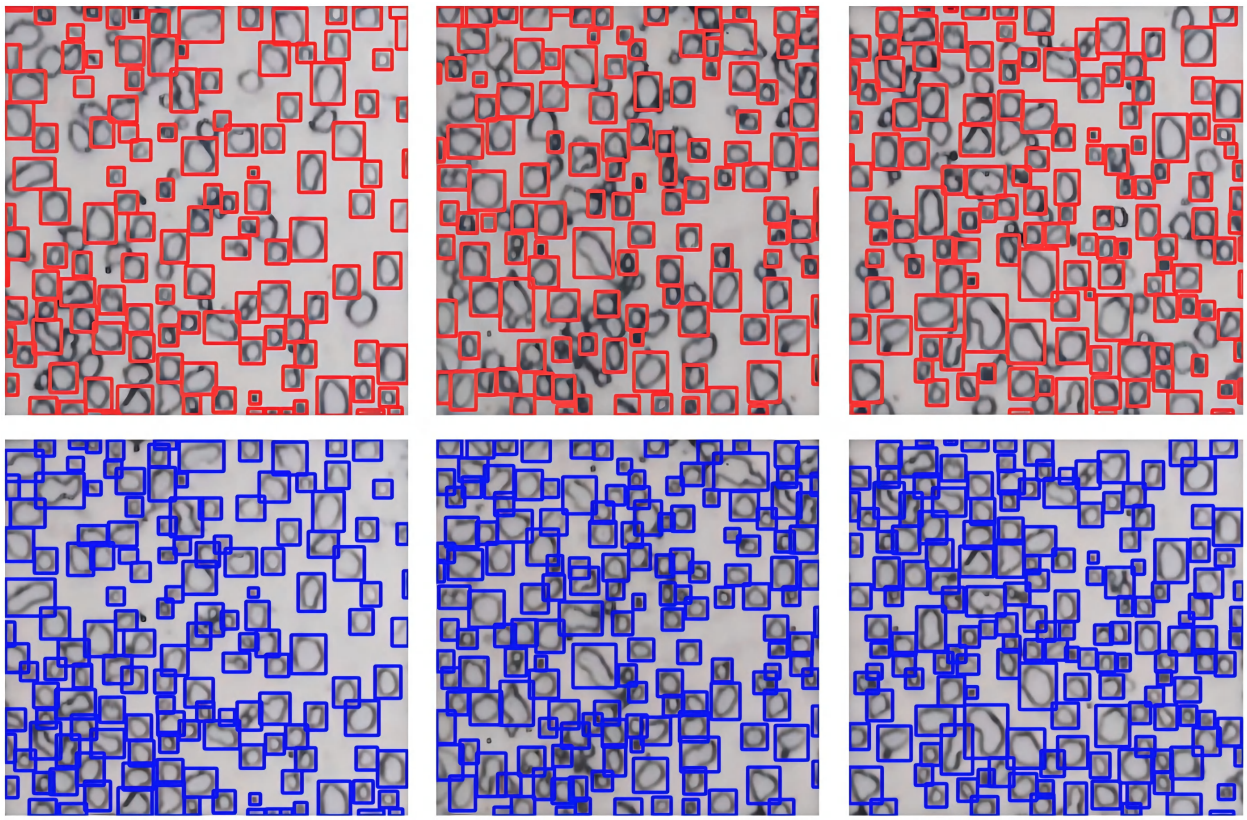

Fig. 6: Results Comparison

Overall, this ablation study clearly illustrates the performance improvements contributed by each module, validates the effectiveness of our proposed improved model, and provides a reference for further optimization research.

F. Experimental Design and Result Analysis of Model Generalization Capability

To further evaluate the adaptability and generalization capability of the proposed model in different task scenarios, we selected the publicly available steel surface defect detection dataset NEU-DET as the benchmark for comparison. This dataset contains six typical types of industrial surface defects, which differ significantly from our self-constructed dataset in terms of defect categories, image texture, and background noise. Therefore, it serves as an effective platform to test the detection performance of models across different data domains.

In this experiment, our model and several mainstream object detection algorithms (including YOLOv8, YOLOv9, YOLOv10, YOLOv11 and yolov12) were independently trained and evaluated on the NEU-DET dataset to ensure fairness. Evaluation metrics include Precision, Recall, and mAP, and the results are shown in Table 2. Our model achieved a high precision of 83.8%, matched YOLOv11 in recall (99%), and attained a nearly optimal mAP of 79.7%, outperforming YOLOv9 and YOLOv10. These results demonstrate the proposed model's strong generalization ability and robustness across different scenarios.

TABLE II: Compare dDifferent categories pairwise

Method	ImageSize	Precision	Recall	mAP
YOLOv8	200*200	80.2	97.0	77.8
YOLOv9	200*200	83.6	96.0	79.3
YOLOv10	200*200	79.0	95.0	74.1
YOLOv11	200*200	85.2	99.0	81.0
YOLOv12	200*200	88.1	97.0	77.6
ours	200*200	83.8	99.0	79.7

G. Random Image Detection

In this spherical cementite dataset, the object detection task is challenging due to the small size of the targets, their large quantity, and the occurrence of target adhesion. The detection results are shown in Figure 6, where the red boxes represent the detections using the baseline model weights, and the blue boxes represent the detections using the modified model weights. From the comparison images, it is evident that the improved YOLOv9 model achieves more precise bounding box localization in complex backgrounds and dense target scenarios. The original YOLOv9 model exhibited missed detections in some images, and the boundary localization was inaccurate in three images. These results indicate that the improved YOLOv9 model significantly enhances detection performance on this dataset, effectively reducing missed detections. Overall, the detection accuracy has been greatly

TABLE III: Ablation experiments

	C3_CD_CBAM	CARAFE	SimAM	Precision/%	Recall/%	mAP
YOLOv9	-	-	-	91.4	87.0	88.7
YOLOv9	√	-	-	89.5	87.0	89.0
YOLOv9	-	√	-	89.8	88.0	89.3
YOLOv9	-	-	√	82.7	87.0	89.1
YOLOv9	√	√	-	91.9	88.0	89.7
YOLOv9	√	√	√	92.6	88.0	89.9

improved, validating the effectiveness of our model improvements.

V. Conclusion

This paper constructs a spherical cementite dataset with rich samples and proposes an improved algorithm based on YOLOv9, specifically optimized for small object detection tasks in metallographic images. By incorporating the C3_CD_CBAM module, CARAFE module, and SimAM module, we have developed a model more suitable for spherical cementite detection on the basis of YOLOv9. These modules integrate advanced techniques such as feature extraction, feature grouping, multi-level feature fusion, and contextual information processing, significantly enhancing the model's performance in complex backgrounds and small object detection. Experimental results on the dataset demonstrate a remarkable improvement in detection accuracy, reaching 89.9%, and achieving precise identification of spherical cementite. This not only promotes the development of metallographic analysis technology but also showcases the broad application potential of deep learning in the field of materials science.

References

[1] M. Umemoto and H. Ohtsuka, "Mechanical properties of cementite," *TETSU TO HAGANE-JOURNAL OF THE IRON AND STEEL INSTITUTE OF JAPAN*, vol. 107, no. 4, pp. 269–289, 2021.

[2] H. Wang, F. Wang, D. Qian, F. Chen, Z. Dong, and L. Hua, "Investigation of damage mechanisms related to microstructural features of ferrite-cementite steels via experiments and multiscale simulations," *International Journal of Plasticity*, vol. 170, p. 103745, 2023.

[3] M. Pinson, H. Springer, T. Depover, and K. Verbeken, "The role of cementite on the hydrogen embrittlement mechanism in martensitic medium-carbon steels," *Materials Science and Engineering: A*, vol. 859, p. 144204, 2022.

[4] T. Yasuda and N. Nakada, "Effect of carbon concentration in austenite on cementite morphology in pearlite," *ISIJ International*, vol. 61, no. 1, pp. 372–379, 2021.

[5] A. S. Koraboyevna, "General information on metallographic analysis," *Ethiopian International Journal of Multidisciplinary Research*, vol. 11, no. 04, pp. 294–300, 2024.

[6] Y. Sun, Y. Zhang, Z. Wei, and J. Zhou, "A classification and location of surface defects method in hot rolled steel strips based on yolov7," *Metalurgija*, vol. 62, no. 2, pp. 240–242, 2023.

[7] W. Teng, Y. Zhang, H. Zhang, and D. Gao, "Surface defect detection of steel based on improved yolov7 model," *Metalurgija*, vol. 63, no. 3-4, pp. 399–402, 2024.

[8] Y. Xiao, Z. Tian, J. Yu, Y. Zhang, S. Liu, S. Du, and X. Lan, "A review of object detection based on deep learning," *Multimedia Tools and Applications*, vol. 79, pp. 23729–23791, 2020.

[9] D. Bhatt, C. Patel, H. Talsania, J. Patel, R. Vaghela, S. Pandya, K. Modi, and H. Ghayvat, "Cnn variants for computer vision: History, architecture, application, challenges and future scope," *Electronics*, vol. 10, no. 20, p. 2470, 2021.

[10] X. Zhao, L. Wang, Y. Zhang, X. Han, M. Deveci, and M. Parmar, "A review of convolutional neural networks in computer vision," *Artificial Intelligence Review*, vol. 57, no. 4, p. 99, 2024.

[11] F. Xin, H. Zhang, and H. Pan, "Hybrid dilated multilayer faster rcnn for object detection," *The Visual Computer*, vol. 40, no. 1, pp. 393–406, 2024.

[12] J. Pan and Y. Zhang, "Small object detection in aerial drone imagery based on yolov8," *IAENG International Journal of Computer Science*, vol. 51, no. 9, pp. 1346–1354, 2024.

[13] W.-H. Wu, J.-C. Lee, and Y.-M. Wang, "A study of defect detection techniques for metallographic images," *Sensors*, vol. 20, no. 19, p. 5593, 2020.

[14] J. Luengo, R. Moreno, I. Sevillano, D. Charte, A. Pelaez-Vegas, M. Fernandez-Moreno, P. Mesejo, and F. Herrera, "A tutorial on the segmentation of metallographic images: Taxonomy, new metaldam dataset, deep learning-based ensemble model, experimental analysis and challenges," *Information Fusion*, vol. 78, pp. 232–253, 2022.

[15] C.-Y. Wang, I.-H. Yeh, and H.-Y. Mark Liao, "Yolov9: Learning what you want to learn using programmable gradient information," in *European conference on computer vision*, pp. 1–21, Springer, 2024.

[16] P. Jiang, D. Ergu, F. Liu, Y. Cai, and B. Ma, "A review of yolo algorithm developments," *Procedia computer science*, vol. 199, pp. 1066–1073, 2022.

[17] A. Vijayakumar and S. Vairavasundaram, "Yolo-based object detection models: A review and its applications," *Multimedia Tools and Applications*, vol. 83, no. 35, pp. 83535–83574, 2024.

[18] M. Ali and Z. Zhang, "The yolo framework: A comprehensive review of evolution, applications, and benchmarks in object detection. computers 2024, 13, 336," 2024.

[19] R. Sapkota, Z. Meng, M. Churuvija, X. Du, Z. Ma, and M. Karkee, "Comprehensive performance evaluation of yolo11, yolov10, yolov9 and yolov8 on detecting and counting fruitlet in complex orchard environments," *arXiv preprint arXiv:2407.12040*, 2024.

[20] J. Bao and X. Yuan, "Yolo-icbam: an improved yolov4 based on cbam for defect detection," in *Fifteenth International Conference on Signal Processing Systems (ICSPS 2023)*, vol. 13091, pp. 520–525, SPIE, 2024.

[21] J. Wang, K. Chen, R. Xu, Z. Liu, C. C. Loy, and D. Lin, "Carafe: Content-aware reassembly of features," in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 3007–3016, 2019.

[22] H. Tang and Y. Jiang, "An improved yolov8n algorithm for object detection with carafe, multiseamhead, and tripleattention mechanisms," in *2024 7th International Conference on Computer Information Science and Application Technology (CISAT)*, pp. 119–122, IEEE, 2024.