electronics

MDPI

Article

Optimized AI Methods for Rapid Crack Detection in Microscopy Images

Ch enxukun Lou [1,2], Lawrence Tinsley [2], Fabian Duarte Martinez [2], Simon Gray [2] and Barmak Honarvar Shakibaei Asli [2,*]

[1] College of Mechanical and Electrical Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China
[2] Centre for Life-Cycle Engineering and Management, Faculty of Engineering and Applied Sciences, Cranfield University, Cranfield, Bedfordshire MK43 0AL, UK
* Correspondence: barmak@cranfield.ac.uk

Abstract: Detecting structural cracks is critical for quality control and maintenance of industrial materials, ensuring their safety and extending service life. This study enhances the automation and accuracy of crack detection in microscopic images using advanced image processing and deep learning techniques, particularly the YOLOv8 model. A comprehensive review of relevant literature was carried out to compare traditional image-processing methods with modern machine-learning approaches. The YOLOv8 model was optimized by incorporating the Wise Intersection over Union (WIoU) loss function and the bidirectional feature pyramid network (BiFPN) technique, achieving precise detection results with mean average precision (mAP@0.5) of 0.895 and a precision rate of 0.859, demonstrating its superiority in detecting fine cracks even in complex and noisy backgrounds. Experimental findings confirmed the model's high accuracy in identifying cracks, even under challenging conditions. Despite these advancements, detecting very small or overlapping cracks in complex backgrounds remains challenging. Our future work will focus on optimizing and extending the model's generalisation capabilities. The findings of this study provide a solid foundation for automatic and rapid crack detection in industrial applications and indicate potential for broader applications across various fields.

Keywords: crack detection; YOLOv8; deep learning; microscopic images; structural integrity

Citation: Lou, C.; Tinsley, L.; Duarte Martinez, F.; Gray, S.; Honarvar Shakibaei Asli, B. Optimized AI Methods for Rapid Crack Detection in Microscopy Images. *Electronics* **2024**, *13*, 4824. https://doi.org/10.3390/electronics13234824

Academic Editors: Jian Sun, Soo-Hyung Kim, Hieyong Jeong and Inseop Na

Received: 14 September 2024
Revised: 16 November 2024
Accepted: 3 December 2024
Published: 6 December 2024

1. Introduction

Accurate crack detection is vital for maintaining the structural integrity of materials across various industrial applications. Traditional methods, such as manual inspection and basic image processing techniques like edge detection and morphological operations, rely heavily on human expertise. Complex backgrounds and noise often challenge these methods, resulting in reduced detection accuracy and efficiency, particularly in the presence of intricate surface details [1]. In contrast, rapid advancements in computer vision and deep learning, particularly convolutional neural networks (CNNs), have enabled significant advancements in feature extraction and pattern recognition, thereby improving detection accuracy and enabling automation [2,3]. However, applying these deep learning techniques to specialized tasks is still limited due to the unique challenges posed by certain image properties, particularly those in scanning electron microscopy (SEM) images. SEM inspections can generate hundreds of images, making manual analysis both time-consuming and error-prone, particularly for untrained or fatigued inspectors who might overlook cracks. To overcome these limitations, our goal is to automate the inspection process through artificial intelligence and ultimately develop a digital twin system capable of real-time, in situ crack detection within SEM images.

Microscopic images, extensively used in materials science, biology, and medicine, pose unique challenges due to their high resolution and substantial noise and artefacts, such as

oxide layers, scratches, and particles [4,5]. These characteristics complicate the detection process, as many existing deep learning models and image processing techniques are not specifically optimized for the detailed and noisy nature of microscopic data [6,7]. As a result, despite advances in crack detection for conventional imaging, a significant research gap remains in addressing the specific challenges posed by microscopic images. Current deep learning models often struggle to generalize to the high complexity and noise levels inherent in microscopic images, resulting in suboptimal detection outcomes in practical applications [8,9].

Recent studies have proposed various methods to enhance crack detection; however, many still struggle to achieve the precision required in complex environments. For example, Pauly et al. [10] employed a deeper neural network to improve road crack detection, achieving greater accuracy and robustness in relatively simple contexts. Building on this, Li et al. [11] introduced SegFormer, a hierarchical Transformer-based network, which outperformed traditional CNN-based models in detecting small cracks on concrete and asphalt surfaces. However, these approaches often face challenges in handling complex backgrounds and detecting fine-scale cracks. To address some of these issues, Zhang et al. [12] proposed an unsupervised learning method combining the Vector Quantized-Variational Autoencoder 2 (VQ-VAE2) and Pixel Sequential Neural Autoregressive Model (PixelSNAIL), enabling crack detection without annotated data. However, it still struggles with the intricate details of small cracks and background noise. Kou et al. [13] developed a YOLOv3-based model for detecting steel strip surface defects, incorporating an anchor-free feature selection mechanism to reduce computation time and dense convolution blocks to enhance feature propagation. While effective across multiple datasets, the model still requires improvement in detecting small-scale cracks and handling complex backgrounds. Hu et al. [14] optimized YOLOv5 for vehicle-mounted pavement crack detection, introducing advanced feature extraction and optimization techniques. Their method achieved high detection accuracy and speed, excelling in complex environments and outperforming traditional and other deep learning models. While these efforts improved detection accuracy, they still exhibited limitations when dealing with small-scale cracks under diverse conditions. Su et al. [15] highlighted the advantages of single-stage detection algorithms like YOLO in applications where precise pixel-level identification is less critical. Despite the advancements with YOLO models in detecting various defects, their effectiveness in addressing the specific challenges of microscopic image analysis, especially in SEM images, remains limited. This underscores the need for more tailored approaches capable of reliably detecting and measuring cracks in such complex scenarios, a goal that this study aims to achieve. Table 1 shows the research contents and results of the above studies.

Table 1. Comparison of different studies on object recognition and classification algorithms.

Study	Approach	Dataset Type	Results	Challenges/Limitations
Pauly et al. [10]	Deep Neural Network for road crack detection	Road Images	Achieved improved detection accuracy and robustness in controlled environments	Struggles with precision in complex environments
Li et al. [11]	SegFormer for small crack detection on concrete/asphalt surfaces	Concrete/Asphalt Surfaces	The recall rate of SegFormer-B5 is 77.4%, which is 4.0% higher than DeepLabV3-R101, and the accuracy rate can reach 98.8%.	Challenges in handling fine-scale cracks in complex backgrounds

Table 1. *Cont.*

Study	Approach	Dataset Type	Results	Challenges/Limitations
Zhang et al. [12]	VQ-VAE2 + PixelSNAIL for unsupervised crack detection	Unsupervised Datasets	The accuracy rate can reach 93.33%	Struggled with detailed features in noisy environments
Kou et al. [13]	YOLOv3 optimized for steel strip defects	Steel Strip Surface Datasets	The proposed model yielded 71.3% mAP on the GC10-DET dataset, and 72.2% mAP on the NEU-DET dataset.	Limitations in detecting small-scale defects in complex conditions
Hu et al. [14]	YOLOv5 optimized for vehicle-mounted pavement crack detection	Vehicle-Mounted Pavement Images	ACompared with YOLOv5, mAP increased by 3.41% and F_1-score increased by 2.25%	Needs improvement for finer detection in more intricate scenarios
Su et al. [15]	Maintaining the Original Dimension-YOLO	Various Defect Datasets	The accuracy rate reached 91.1%, the number of parameters was reduced by 19.7%, and the computational complexity was reduced by 35.9%.	The model is relatively large

Addressing the gaps identified in the current literature, this study optimizes the YOLOv8 model for crack detection in microscopic images, specifically in SEM images. YOLOv8 introduces several key advancements that significantly enhance its performance and versatility, establishing it as a state-of-the-art model in object detection. The model is available in multiple versions—N, S, M, L, and X—each tailored to different deployment platforms and application scenarios, ensuring scalability and adaptability [16,17].

The YOLOv8 architecture was systematically refined to enhance both computational efficiency and detection accuracy. In the backbone, it adopts the Cross Stage Partial (CSP) strategy, which improves efficiency by dividing the feature map into two branches—one processed through convolutional layers and the other bypassing them—thereby reducing redundant computations. The C3 module, originally used in previous versions, implements a CSP bottleneck with three convolutional layers. However, in YOLOv8, this was replaced by the C2f module, a faster and optimized implementation of the C2 module (CSP bottleneck with two convolutions). The C2f module achieves improved execution speed while maintaining comparable performance by streamlining the internal operations of the bottleneck structure [18–20].

Furthermore, the backbone integrates the Spatial Pyramid Pooling Fast (SPPF) module, which compresses spatial dimensions efficiently while preserving critical multi-scale contextual information, reducing computational overhead without sacrificing performance. In the neck, YOLOv8 continues to employ the Path Aggregation Network (PANet), with further structural optimizations to enhance the fusion of high-level semantic features with low-level spatial information. This design enables the model to effectively capture fine-grained details and global context, improving detection accuracy across objects of varying scales and complexities. In the head of the network, the transition from an Anchor-Based to an Anchor-Free detection paradigm simplifies model design and improves its robustness in detecting objects of varying sizes and scales [21,22]. The model framework of YOLOv8 is shown in Figure 1.

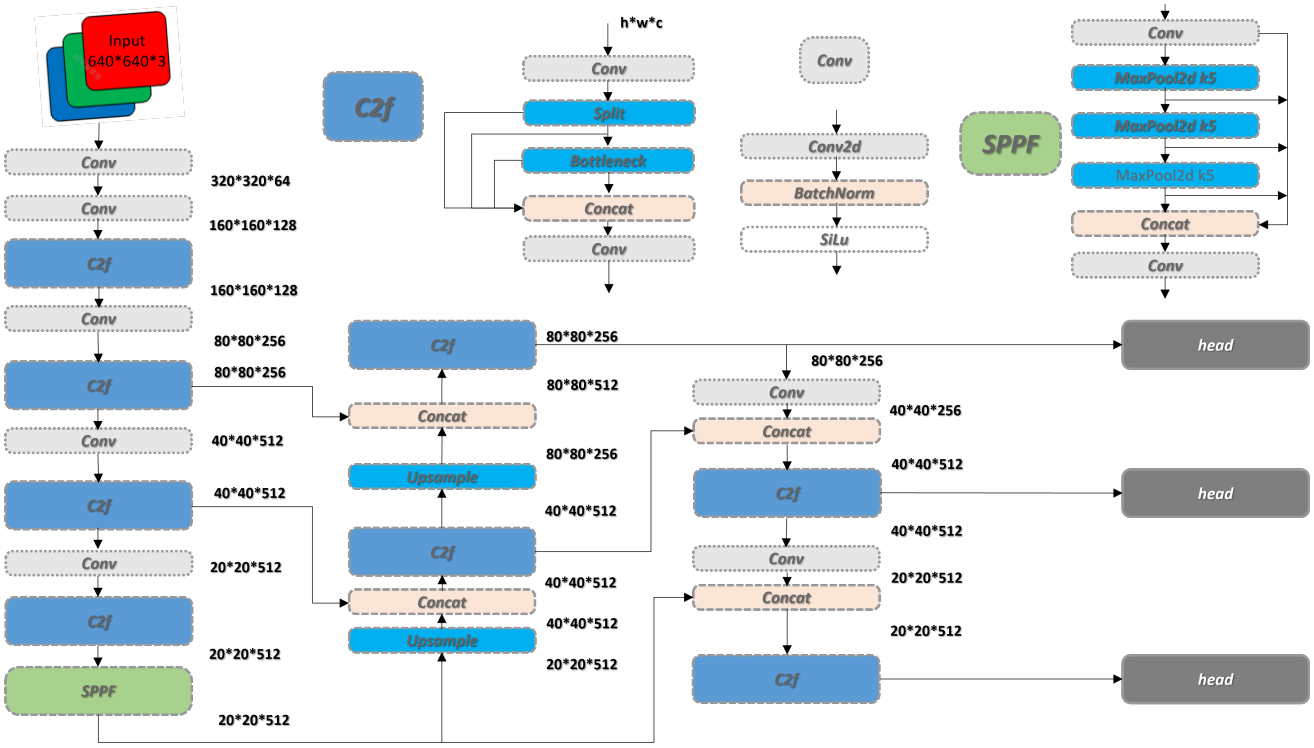

Figure 1. YOLOv8 model framework.

YOLOv8 integrates advanced loss functions such as Varifocal Loss (VFL) for classification and Distance-Focal L1 (DFL) Loss for regression to further optimize the learning process. These loss functions enhance detection confidence and localization precision, which are critical for complex object detection tasks. Additionally, YOLOv8 introduces a Task-Aligned Assigner strategy, replacing traditional IoU-based sample allocation methods. This strategy dynamically aligns positive and negative samples based on prediction quality, facilitating more balanced and efficient model training [23–25].

These enhancements make YOLOv8 more adaptable and robust for a wide range of object detection applications, significantly improving both computational efficiency and detection performance compared to its predecessors. Building on this foundation, this study integrates advanced optimization strategies, including WIoU and BiFPN, which further refine the model's localization accuracy and robustness. These improvements are particularly critical at lower confidence thresholds, which are essential for detecting fine cracks and defects in challenging environments. WIoU is an advanced loss function that enhances bounding box regression by incorporating an attention-based mechanism, adjusting weights based on the centre distance and aspect ratios of bounding boxes. BiFPN, on the other hand, is a feature fusion network that enhances detection performance by efficiently combining features from different levels in a bidirectional manner, improving both high-level semantic information and low-level detail extraction. The baseline YOLOv8 model initially demonstrated promising performance, achieving an F_1-score of 0.84 and a mean average precision (mAP@0.5) of 0.876, with a distance measurement error within 3%, indicating strong potential for accurate crack detection. Through independent optimizations, WIoU increased the F_1-score to 0.85 and the mAP@0.5 to 0.895. Similarly, BiFPN maintained a comparable mAP@0.5 of 0.894, with enhanced stability, demonstrating the effectiveness of these enhancements in significantly boosting the YOLOv8 model's capability for precise and rapid automated crack detection in SEM images.

2. Methods

This study focuses on optimizing the model through structural modifications and parameter fine-tuning. By adjusting the YOLO framework and incorporating advanced

components, such as re-parametrized networks and various loss functions, the goal is to enhance detection accuracy and efficiency.

2.1. Parameter Adjustment and Optimization

Researching and refining algorithms is a critical aspect of this study. Based on a comprehensive literature review, several potential modifications and enhancements to the framework were identified, organized as follows:

- ☐ Make slight modifications to the framework.
- ☐ Add additional detection layers.
- ☐ Integrate attention mechanisms, such as Convolutional Block Attention Module (CBAM) [26], Squeeze and Excitation (SE) [27], and Self-Attention (SA) [28].
- ☐ Employ alternative loss functions, including Distance-IoU (DIoU) [29], Generalized-IoU (GIoU) [30], and Scalable-IoU (SIoU) [31]. These functions can more effectively measure the difference between the predicted and actual bounding boxes, thereby improving the model's positioning accuracy.
- ☐ Utilize re-parametrized networks, such as Re-parametrized VGG (RepVGG) [32].

This article mainly increases the detection accuracy by slightly modifying the YOLO framework and using other loss functions to improve the measurement accuracy.

2.1.1. Loss Function Improvement

The IoU loss, or Intersection over Union loss, is a commonly used loss function in object detection tasks. In the object detection task, IoU is used to measure the overlap between the anchor box and the target box. It effectively masks the interference of the bounding box size in the form of a ratio, so that the model can well balance the learning of large and small objects when using $1 - IoU$ ($1 - IoU$ represents the difference between the real box and the predicted box) as the BBR loss [33,34]. The function of IoU loss is defined as

$$IoU = \frac{A \cap B}{A \cup B}, \tag{1}$$

$$\mathcal{L}_{IoU} = 1 - IoU, \tag{2}$$

where the symbols A and B represent the areas of two regions, respectively. Here, $A \cap B$ denotes the intersection area of regions A and B, which is the overlapping part between the two regions. Conversely, $A \cup B$ represents the union area of regions A and B, which includes all parts of both regions. $\mathcal{L}$ denotes "Loss", and $\mathcal{L}_{IoU}$ indicates that this is an IoU-based loss function.

YOLOV8 uses Complete Intersection over Union (CIoU), which is more robust and sensitive in capturing target shape, improving positioning accuracy and handling complex scenes. Its definition is as follows:

$$\mathcal{L}_{CIoU} = \mathcal{L}_{IoU} + \frac{\rho^2(b, b^{gt})}{c^2} + \alpha v, \tag{3}$$

where IoU measures the overlap between the predicted box b and the ground truth b^{gt}. $\rho^2(b, b^{gt})$ is the centre point distance between b and b^{gt}, c is the diagonal of the smallest enclosing box, and αv accounts for aspect ratio consistency, providing a comprehensive bounding box regression metric. Among them,

$$v = \frac{4}{\pi^2}(\arctan\frac{w^{gt}}{h^{gt}} - \arctan\frac{w}{h})^2, \tag{4}$$

where w and h are the width and height of the predicted box, respectively, with w^{gt} and h^{gt} being ground truths. The arctan function captures the angle of the aspect ratios, and the formula penalizes deviations, improving shape alignment.

$$\alpha = \frac{v}{\mathcal{L}_{IoU} + v}, \tag{5}$$

α is a balanceparameter (this coefficient does not participate in the gradient calculation). Here, the priority is given according to the IoU value. The coefficient is larger when the IoU between the prediction box and the target box is larger.

Although CIoU has many advantages, such as shape invariance, sensitivity to positioning accuracy, and comprehensiveness, it still has some shortcomings. First, CIoU has high computational complexity. Compared with the traditional IoU loss function, CIoU involves more calculation steps, including calculating the full cross-area and introducing correction factors, which may increase the computational burden of model training. Second, some parameters in CIoU (such as the weight parameters of the correction factor) have a great impact on the performance of the loss function and need to be carefully adjusted to obtain the best model performance. In addition, since CIoU considers more information, the model may require more data and longer training time to fully learn this information, which is particularly unfavourable for training small-scale datasets [35–37].

WIoU proposed by Tong et al. [38] significantly reduces regression error and improves model performance by designing an attention-based BBR loss (WIoU v1), and an improved version with monotonic FM (WIoU v2) and dynamic non-monotonic FM (WIoU v3). According to multiple experiments, WIoU has the best effect. Therefore, it was ultimately chosen as the loss function.

WIoU v1 loss definition:

$$\mathcal{L}_{WIoUv1} = \mathcal{R}_{WIoU}\mathcal{L}_{IoU}, \tag{6}$$

$$\mathcal{R}_{WIoU} = \exp\left(\frac{(x - x_{gt})^2 + (y - y_{gt})^2}{(W_g^2 + H_g^2)^*}\right), \tag{7}$$

where (x, y) and (x_{gt}, y_{gt}) are the centre coordinates of the predicted and ground truth boxes, respectively. W_g and H_g represent the width and height of the minimum bounding box, respectively. To prevent R_{WIoU} from generating gradients that hinder convergence, W_g and H_g are separated from the computational graph (the superscript * indicates this operation).

The improvement of WIoU v2 is to reduce the contribution of simple examples to the loss value by designing a monotonic focusing mechanism.

WIoU v2 loss definition:

$$\mathcal{L}_{WIoUv2} = \left(\frac{\mathcal{L}^*_{IoU}}{\overline{\mathcal{L}_{IoU}}}\right)^{\gamma}\mathcal{L}_{WIoUv1}, \tag{8}$$

where $\overline{\mathcal{L}_{IoU}}$ represents a moving average with momentum m. The dynamically updated normalization factor, defined as $r = \left(\frac{\mathcal{L}^*_{IoU}}{\overline{\mathcal{L}_{IoU}}}\right)^{\gamma}$, ensures that the gradient gain remains relatively high, effectively addressing the problem of slow convergence in the later stages of training.

The concept of outlier degree is introduced to quantify the quality of misaligned bounding boxes, defined as

$$\beta = \frac{\mathcal{L}^*_{IoU}}{\overline{\mathcal{L}_{IoU}}}, \quad \beta \in [0, +\infty). \tag{9}$$

a smaller value of β implies a higher penalty on lower-quality bounding boxes, thus directing the boundary regression toward boxes of moderate quality. This helps prevent

harmful gradients caused by low-quality samples. Using the monotonic focusing factor β, the WIoU v3 loss definition is

$$\mathcal{L}_{WIoUv3} = r\mathcal{L}_{WIoUv1}, r = \frac{\beta}{\delta\alpha^{\beta-\delta}}, \tag{10}$$

where δ and α are constants that balance the scaling factor, ensuring effective performance across different scenarios.

The integration of WIoU into the YOLOv8 model requires several targeted modifications to ensure that the new loss function is properly incorporated and improves detection accuracy. The process begins with identifying the section in the model's code responsible for computing the IoU metric, typically found in the logic handling bounding box regression. The standard IoU or CIoU formula in this section is replaced with the WIoU algorithm, which enhances traditional IoU by introducing additional terms that account for the distance between bounding box centres and their aspect ratio consistency. This adjustment refines the precision of the bounding box regression, increasing the model's robustness and adaptability across a variety of object shapes and scales.

Following this modification, the component that computes the model's loss function is updated to ensure the use of WIoU throughout the training process. This involves replacing the original IoU-based loss with the WIoU loss function. The WIoU loss offers better optimization by reducing false positives and false negatives, particularly in complex scenes with overlapping objects or small targets. Fine-tuning WIoU-specific parameters—such as centre distance and aspect ratio weights—is essential during this phase to ensure that the model achieves optimal generalization and stability across diverse datasets and environments.

Once the WIoU modifications are completed, the model configuration files and training scripts are updated to integrate the new loss function into the entire training and validation pipeline. Adjustments are made to relevant parameters, such as confidence thresholds and IoU thresholds, to fully leverage the WIoU loss function's capabilities. The updated model is then rigorously tested and validated, with experimental comparisons conducted to evaluate the improvements over traditional IoU-based losses. Special focus is placed on the model's performance in challenging environments, such as cluttered backgrounds and low confidence scenarios, to ensure reliability and robustness across a broad range of conditions.

2.1.2. Structural Improvements

Since YOLOv8 adopts PANet for feature aggregation, this study builds upon the BiFPN architecture proposed by Tan et al. [39], which introduces a weighted bidirectional feature pyramid network to enhance multi-scale feature fusion. FPN (feature pyramid network), a widely used structure in object detection, combines low-level positional and high-level semantic features through a top-down pathway with lateral connections, generating feature maps that facilitate accurate detection across varying object scales. However, FPN suffers from limited aggregation efficiency and redundant nodes, reducing its overall effectiveness. To overcome these limitations, PANet (Path Aggregation Network) extends FPN by introducing a bottom-up path that enriches feature propagation and improves multi-scale utilization. While PANet enhances the fusion process, it lacks a dynamic mechanism for balancing the importance of input features adaptively.

Figure 2 compares the structures of FPN and BiFPN, highlighting the advantages of the latter. As shown in Figure 2a, the standard FPN relies on a unidirectional top-down pathway, which restricts the effectiveness of multi-scale feature fusion. In contrast, Figure 2b illustrates the BiFPN, where feature maps are aggregated in both top-down and bottom-up directions across multiple repeated layers. This bidirectional design enables more effective feature interaction across scales and improves the model's ability to detect objects of varying sizes, making BiFPN a more efficient solution for complex object detection tasks.

Building on PANet's foundation, BiFPN introduces several innovations to further enhance feature fusion. It employs a bidirectional path, where semantic information flows

downward from high-level features, and positional information flows upward from low-level features. Additional edges are introduced between input and output nodes within the same layer, facilitating feature reuse, while nodes with only one input are removed to simplify the network structure. If input and output nodes reside within the same layer, more connections are added to strengthen feature fusion. Each bidirectional path functions as a feature layer that is reused multiple times, with the optimal number of repetitions determined through a Neural Architecture Search (NAS). Additionally, BiFPN applies a fast normalized fusion mechanism with learnable weights to prioritize the most relevant input features, enhancing both training efficiency and model performance. These innovations allow BiFPN to surpass both FPN and PANet in feature aggregation and overall performance, particularly in demanding object detection scenarios.

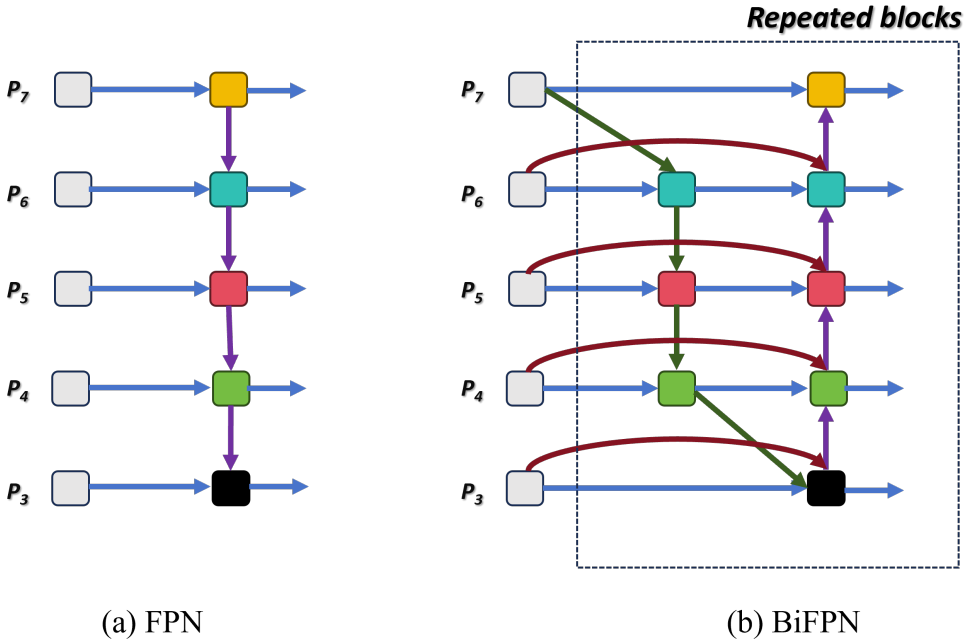

Figure 2. Comparison of FPN and BiFPN structures: (**a**) Feature Pyramid Network (FPN): A traditional top-down architecture for feature fusion, enabling feature maps at different scales to be combined for enhanced multi-scale representation, and (**b**) Bidirectional Feature Pyramid Network (BiFPN): An advanced architecture with bidirectional connections and repeated fusion blocks, designed for more efficient and scalable multi-scale feature aggregation by leveraging both top-down and bottom-up pathways.

The calculation expression is:

$$P_6^{td} = Conv\left(\frac{w_1 \cdot P_6^{in} + w_2 \cdot Resize(P_7^{in})}{w_1 + w_2 + \epsilon}\right), \tag{11}$$

$$P_6^{out} = Conv\left(\frac{w_1' \cdot P_6^{in} + w_2' \cdot P_6^{td} + w_3' \cdot Resize(P_5^{out})}{w_1' + w_2' + w_3' + \epsilon}\right). \tag{12}$$

Here, P_6^{td} and P_6^{out} are the outputs after top-down and bottom-up feature fusion, respectively. The weights w_1, w_2, w_1', w_2', and w_3' determine the importance of each input feature map, and ϵ is a small constant to avoid division by zero. The `Conv` operation applies a convolution to the fused features, enhancing training speed and model performance by differentiating the significance of various input features.

Figure 3 demonstrates the integration of BiFPN into the YOLOv8 architecture. To facilitate this process, the model's configuration must be modified to accommodate for BiFPN being the primary feature fusion module, replacing the original FPN. This adjustment

requires fine-tuning key parameters, including the number of feature levels and the repetition count, to fully exploit BiFPN's multi-scale fusion capability. These modifications are essential for improving the fusion of features across various object sizes and spatial resolutions, thereby enhancing the model's overall detection performance.

The BiFPN module is subsequently embedded into the feature extraction pipeline, introducing a bidirectional fusion mechanism that ensures seamless information exchange between high-level semantic features and low-level spatial details. The bidirectional flow allows the model to effectively capture both global and local information. A learnable weighting mechanism further optimizes the fusion process by dynamically assigning importance to different feature levels, enabling the efficient aggregation of informative features. The BiFPN design ensures full compatibility with the backbone network and detection head, maintaining smooth integration within the YOLOv8 framework without disrupting the detection pipeline.

To ensure the consistent application of BiFPN during both the training and inference phases, the task coordination module is revised. This involves updating function calls and adjusting relevant parameters to replace references to the original feature pyramid network with the integrated BiFPN. These revisions guarantee the effective utilization of bidirectional feature fusion throughout the detection process, improving both the robustness and precision of the model. The performance improvements resulting from this integration are systematically evaluated through extensive training and validation, demonstrating the superiority of BiFPN-enhanced YOLOv8 in multi-scale object detection scenarios.

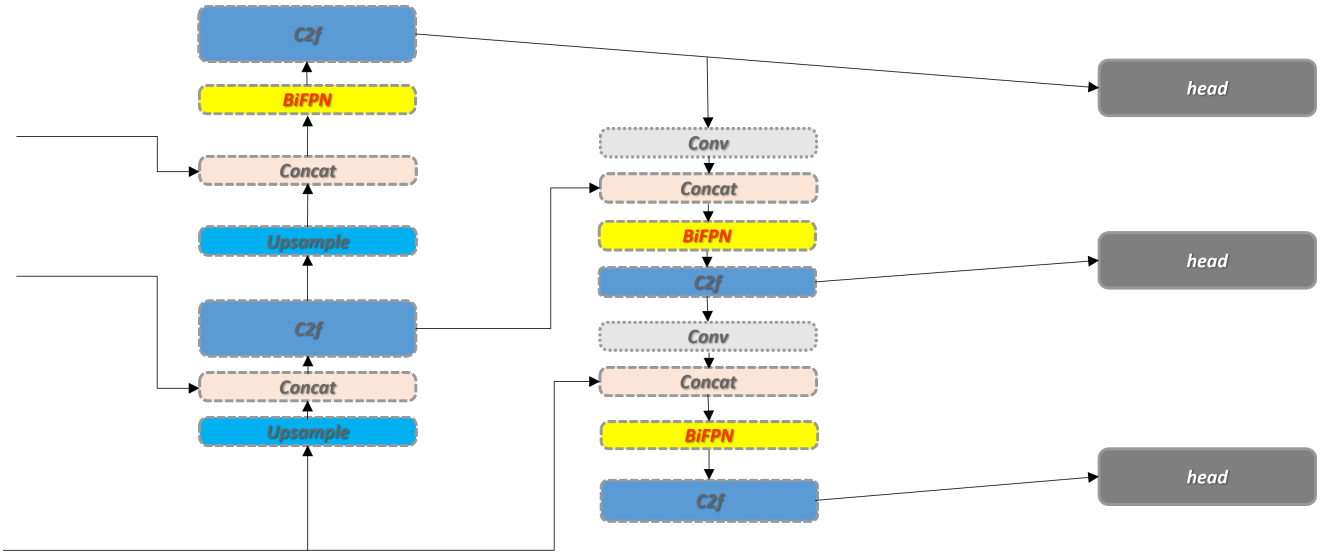

Figure 3. BiFPN integration framework within the YOLOv8 architecture.

2.2. Automatic Scale Recognition and Length Measurement

This section outlines the process for automatically measuring crack lengths in microscopic images using a pre-trained YOLOv8 model. The first essential step involves the recognition of the scale marker embedded within the SEM images, which provides a reference for calculating physical dimensions. These scale markers, commonly labelled with physical measurements (e.g., "200 μm"), establish the pixel-to-length ratio necessary for an accurate conversion from pixel units to real-world dimensions. The YOLOv8 model is trained to detect both scale markers and crack regions within the same image. During detection, the model identifies relevant features, including the scale marker and cracks. If the scale marker is not detected initially, contrast enhancement techniques such as Contrast Limited Adaptive Histogram Equalization (CLAHE) are applied to improve image visibility. This step increases contrast, making the marker easier to detect, after which the detection process is repeated to ensure all features are identified.

After the scale is identified, the crack detection algorithm is used to identify and measure the crack. Cracks are detected by extracting the coordinates and size of the crack. The actual length of the scale is then converted to pixel length to calculate the actual length of the crack. This method can accurately and automatically measure the crack length in microscopic images, laying a solid foundation for further analysis and evaluation. The formula used is

$$\text{Crack Length} = \left(\frac{\text{Crack Pixel Length}}{\text{Scale Pixel Length}}\right)(\text{Scale Actual Length}). \tag{13}$$

For example, if the actual length of the scale is 200 µm and it is detected as 100 pixels in the image, each pixel would correspond to 2 µm. If a detected crack measures 50 pixels, its actual length would be 100 µm. This method allows for accurate measurements of crack lengths in microscopic images, as demonstrated in Figure 4. In Figure 4, multiple cracks are detected, and their lengths are marked with red bounding boxes. Each labelled length corresponds to the pixel-based scale, ensuring precise and reliable crack measurements for further analysis.

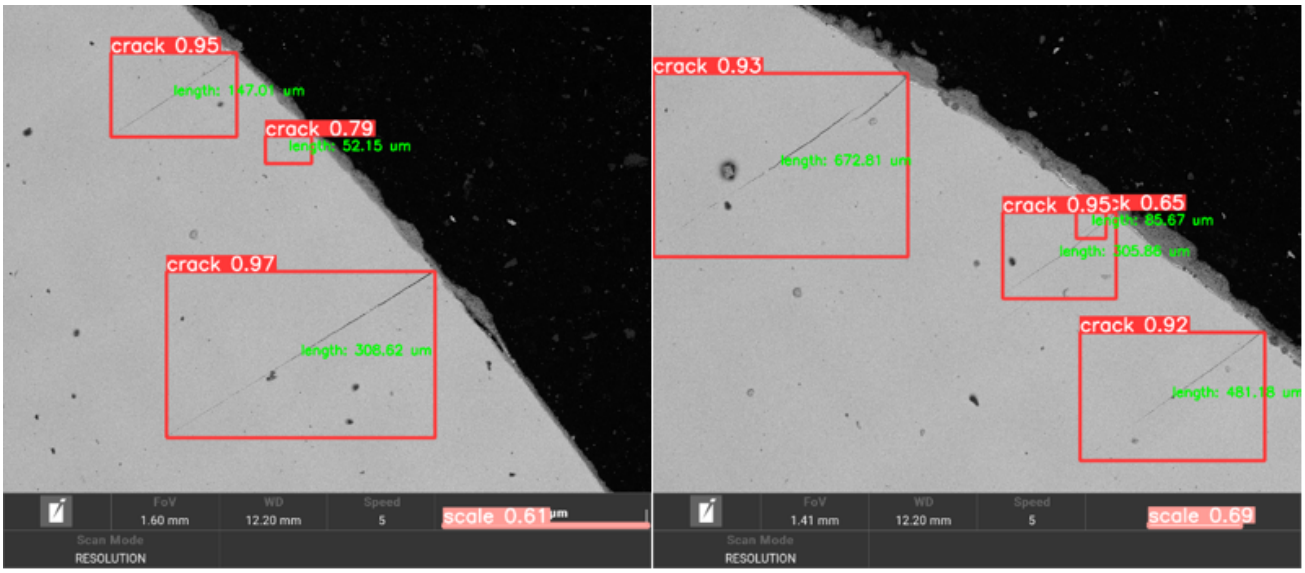

Figure 4. Crack length measurement performed with YOLOv8.

3. Experimental Procedure

Before delving into the experimental design and methodology, it is essential to clarify the objectives and challenges this experiment aims to address. The core aim is to develop and train a YOLOv8 model that can precisely detect cracks in microscopic images. The experiment was thoughtfully laid out to achieve a robust and practical model, covering everything from data collection to model evaluation. The following sections will walk through each step in detail, explaining why they matter for this research.

3.1. Experimental Design

To ensure the robustness and generalizability of the model, we specifically incorporated a diverse set of images sourced from various SEM devices and multiple sample types. This diversity allowed the model to adapt to a broader range of imaging conditions, reducing the risk of overfitting to any particular device or material type. The model's performance was further validated through testing on two distinct SEM image datasets, confirming its ability to deliver consistent detection results across different imaging platforms and demonstrating its effectiveness in generalizing to diverse environments.

To comprehensively evaluate the crack detection performance of the YOLOv8 model in microscopic images, the experimental procedure was structured into several key steps, allowing for a thorough assessment at each stage of the process.

- Dataset creation and labelling: The dataset creation process involves compiling a high-resolution microscopic image dataset, with each image carefully labelled to indicate the presence of cracks and their respective characteristics. Detailed labelling standards were established to ensure accuracy, particularly for complex cases such as overlapping cracks. These cases were addressed by implementing a hierarchical labelling strategy, differentiating cracks based on depth, orientation, and intensity to capture intricate crack patterns.
- Data enhancement and preprocessing: Various data augmentation techniques were applied to enrich the dataset and improve the model's generalization capability. These techniques included random rotations (0–360 degrees), flipping, scaling (up to 20%), brightness adjustments, and the addition of Gaussian noise, simulating diverse imaging conditions the model might encounter. Preprocessing also included noise reduction methods such as Gaussian and median filtering to enhance image quality, thereby ensuring clearer feature extraction and more accurate learning during the training phase.
- Dataset splitting: The dataset was divided into training, validation, and test sets with a 60-20-20 ratio. This balanced split ensured a representative distribution of both simple and complex crack formations across all sets, minimizing the risk of bias. The stratified splitting technique was used to maintain a uniform distribution of cracks in each subset, thus enhancing the model's ability to generalize across different crack types during training, validation, and testing.
- Model initialization and parameter tuning: The YOLOv8 model was initialized with pre-trained weights to leverage transfer learning. Key parameters such as learning rate, batch size, and number of epochs were tuned to optimize performance. A grid search method was employed to find the optimal learning rate, which was finalized at 0.001 after evaluating validation loss across a range of values between 0.0001 and 0.01. The batch size was set to 16, balancing computational efficiency and model performance, while the number of epochs was capped at 150, with early stopping implemented to terminate training when validation loss ceased to improve after 10 consecutive epochs. Momentum and weight decay were also adjusted to prevent overfitting and stabilize convergence.
- Model training and performance optimization: The training process focused on refining the model's performance through targeted optimizations. A learning rate scheduler was employed to dynamically reduce the learning rate when validation loss plateaued, ensuring smoother convergence. Additionally, early stopping was utilized to prevent excessive training when no further improvements in validation loss were observed. Data augmentation techniques were applied throughout the training phase to improve generalization by exposing the model to varied input conditions.
- Model evaluation: The model's accuracy and robustness are evaluated on the validation set using key performance indicators. Additionally, a comparative analysis of models before and after optimization is conducted to identify the most effective solution.

3.2. Evaluation Indicators

This section covers the metrics used to evaluate the model's performance: precision, recall, F_1-score, and mean average precision (mAP). The calculation methods for each metric, as well as their significance in the context of crack detection, are detailed below:

Precision: Measures the accuracy of the positive predictions. It is calculated as

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}. \tag{14}$$

High precision indicates a low false positive rate.

Recall: Measures the ability of the model to identify all relevant instances. It is calculated as

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}. \quad (15)$$

High recall indicates a low false-negative rate.

F_1-score: The harmonic means of precision and recall, providing a balance between the two. It is calculated as

$$F_1\text{-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision+Recall}}. \quad (16)$$

It is important to note the relationship between the F_1-score and the confidence threshold. In object detection models, the confidence threshold determines whether a predicted bounding box is considered positive. As the confidence threshold increases, the model tends to predict fewer positive instances, improving precision but potentially reducing recall. Conversely, lowering the threshold can increase recall at the cost of more false positives, thereby decreasing precision. The F_1-score reflects the balance between these two metrics under the given confidence threshold. Thus, choosing an optimal confidence threshold is crucial for maximizing the model's overall performance, particularly in complex crack detection scenarios.

mAP: Measures the precision across all recall levels. It is a comprehensive metric that evaluates the model's performance in detecting objects at various thresholds.

4. Experimental Results

To make sense of the initial experimental results, it is crucial to first clarify the conditions and methods behind them. Evaluating how the YOLOv8 model performs helps us see how well it detects cracks in SEM images. The dataset was carefully prepared and processed to reflect real-world scenarios as closely as possible. In the next section, we will look at what the results show—the strengths of the model, and where there is room for improvement.

4.1. Preliminary Experimental Results

To understand the preliminary experimental results, it is first necessary to provide a detailed explanation of the data augmentation techniques and dataset proportioning methods utilized. This study employed augmentation strategies such as rotation, scaling, and contrast adjustment to enhance the dataset's variability, ensuring that it more accurately reflects real-world scenarios. The dataset proportions were carefully designed to achieve a balanced distribution of crack types and sizes, thereby improving the model's robustness and generalization capabilities. The dataset was divided into training, validation, and test sets in a 60-20-20 ratio to ensure balanced data distribution across the training, validation, and testing phases. These approaches are critical in enhancing the model's ability to accurately detect cracks in SEM images. This section will discuss the results in detail, highlighting the model's strengths and identifying areas for further optimization.

As shown in Figure 5, the F_1–confidence curve indicates that the model achieves an optimal F_1-score of 0.84 at a confidence threshold of 0.367, effectively balancing precision and recall. However, the "crack" class consistently underperforms compared to the "scale" class, suggesting difficulties in detecting smaller or less distinct features. Figure 6's precision–recall curve further illustrates this disparity, with the "scale" class achieving a high precision of 0.953, while the "crack" class only reaches 0.800, indicating the model's tendency to miss finer details in cracks while maintaining high specificity for scales. Although the model's mAP at IoU = 0.50 (mAP@0.5) is robust at 0.876, the lower mAP at IoU thresholds ranging from 0.50 to 0.95 (mAP50-95) of 0.534 indicates challenges in consistently detecting smaller or more complex cracks across varying IoU thresholds. In Figure 7, the loss function comprises three key components: localization loss (box_loss), distribution focal loss (dfl_loss), and classification loss (cls_loss). The localization loss quantifies

the discrepancy between the predicted bounding box and the ground truth using CIoU, with smaller values indicating more precise localization. Distribution focal loss refines the prediction of bounding box boundaries by assigning greater weight to uncertain or challenging samples, improving the precision of coordinate regression. The classification loss measures the consistency between the predicted class and the ground truth label, with lower values indicating higher classification accuracy. As depicted in Figure 7, the training and validation losses converge smoothly, demonstrating stable learning, yet the fluctuations in precision and recall curves suggest sensitivity to certain sample variations. This highlights the need for further optimization to improve recall and detection stability, particularly in complex backgrounds or for low-confidence instances.

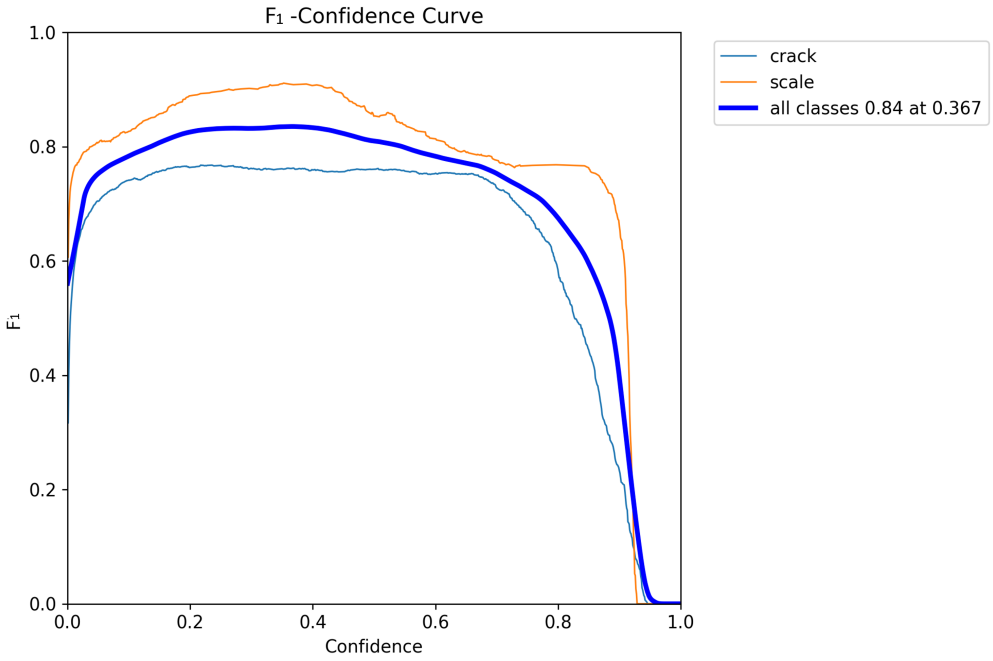

Figure 5. F_1–confidence curve (original YOLOv8).

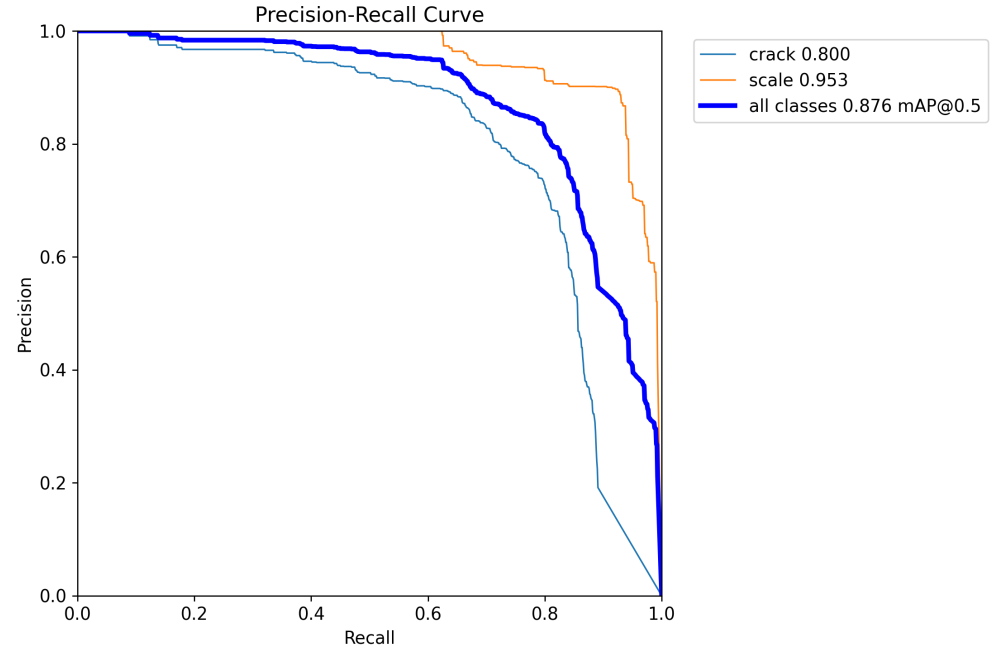

Figure 6. Precision–recall curve (original YOLOv8).

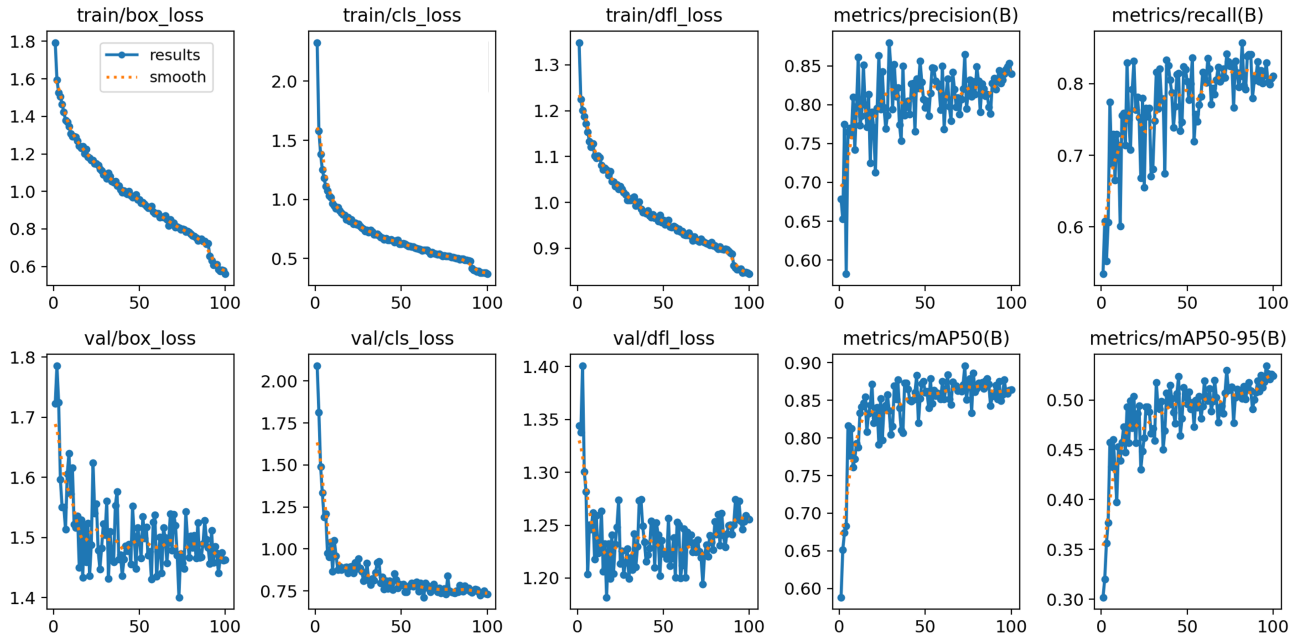

Figure 7. Comprehensive analysis of training and validation loss curves (left 3 columns), including precision, recall, and mean average precision, at various thresholds for model evaluation (right 2 columns).

As shown in Figure 8, the model exhibits strong performance in detecting cracks, accurately identifying distinct and well-defined cracks across different surfaces. The quantitative results emphasize the model's reliability in handling various surface textures and crack patterns, supporting precise defect measurement and analysis under microscopic conditions.

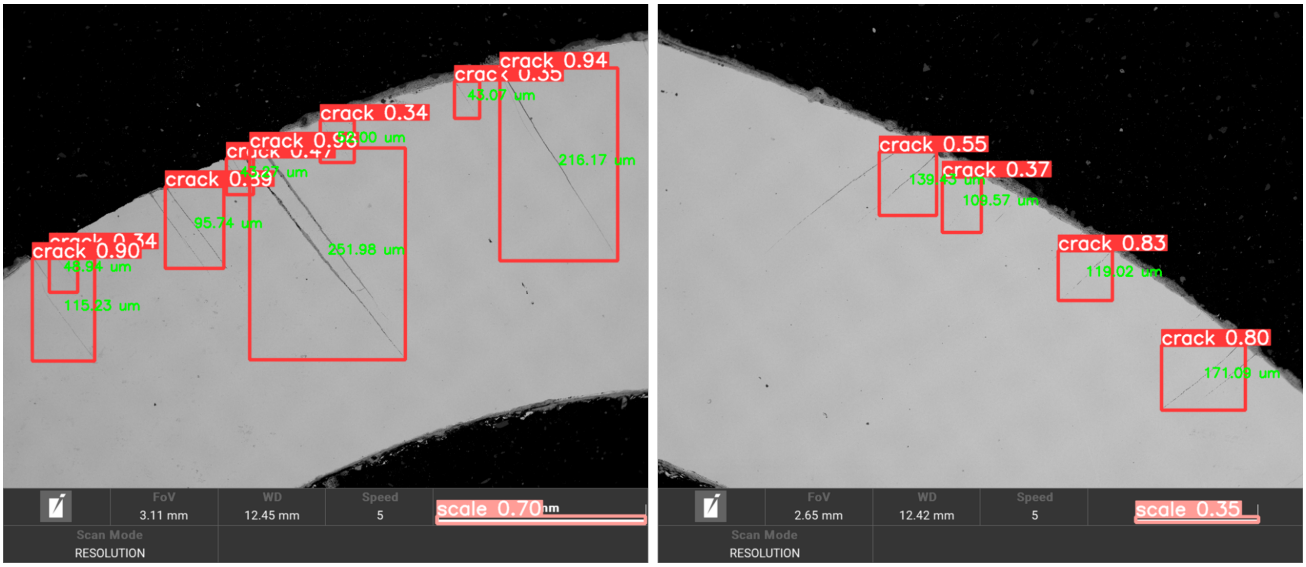

Figure 8. Quantitative results of crack detection and prediction in microscopy.

Figure 9 is a comparison of the results of manual distance measurement and model prediction. The data comparison in the figure is placed in Table 2. As can be seen from Table 2, the error of distance measurement is within 3%. Considering that manual measurements have relative errors, the predicted distance in this experiment has a high accuracy rate.

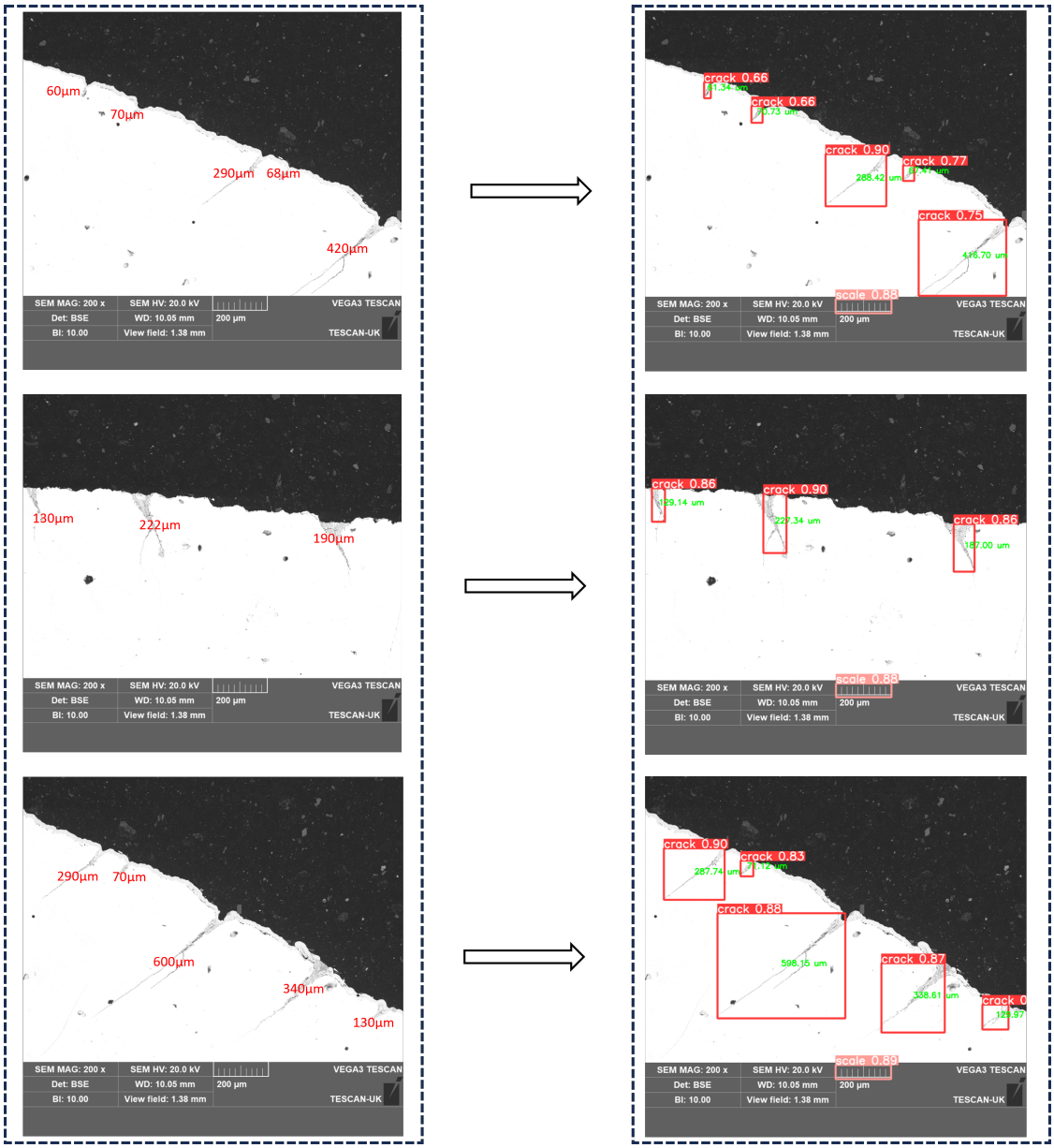

Figure 9. Comparison between manual distance measurement and prediction results.

Table 2. Comparison of length measurement results including model prediction and error probability.

Manual Distance Measurement (μm)	Model Prediction Results (μm)	Error Probability (%)
60	61.34	2.23
70	70.73	1.04
290	288.42	0.55
68	67.41	0.87
420	416.7	0.79
130	129.14	0.66

Table 2. *Cont.*

Manual Distance Measurement (μm)	Model Prediction Results (μm)	Error Probability (%)
222	227.34	2.24
190	187	1.58
290	287.74	0.78
70	71.12	1.6
600	598.15	0.31
340	338.61	0.41
130	129.97	0.02

However, challenges remain in detecting overlapping cracks or those with unclear features, which often present lower contrast and less distinct boundaries, complicating their detection process. This performance limitation can be attributed to the model's inherent difficulty in differentiating subtle variations in crack patterns, leading to missed detections or false negatives. Furthermore, as shown in Figure 10, the initial model exhibits inaccuracies in identifying cracks within the images, frequently missing less obvious or overlapping cracks. This shortcoming highlights the need for further optimization to enhance the model's ability to accurately detect all types of cracks, ensuring comprehensive and reliable defect identification.

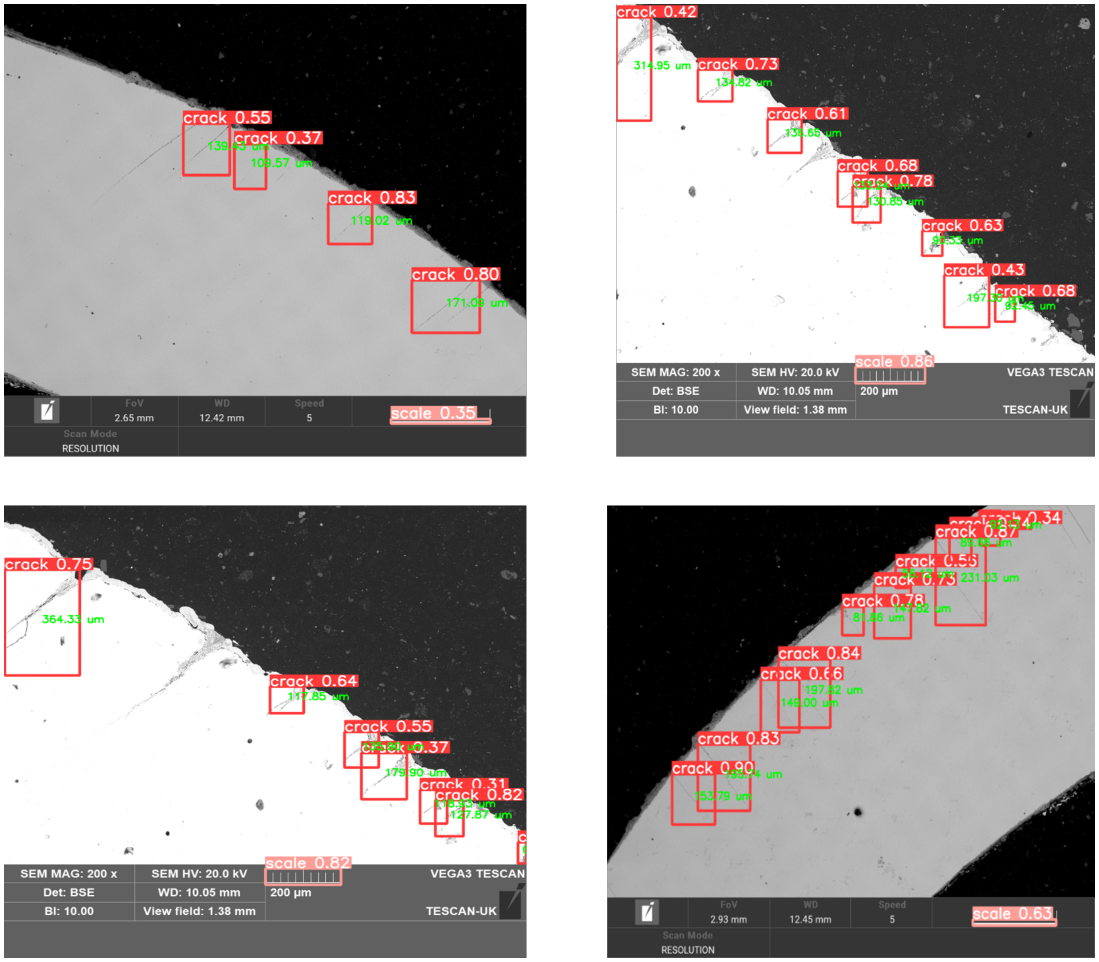

Figure 10. Evaluation of crack detection performance in microscopy with YOLOv8.

4.2. Experimental Results After Optimization

To effectively showcase the impact of the optimizations, it is important to briefly compare the model's performance before and after implementing WIoU and BiFPN. The analysis highlights improvements in key metrics such as F_1-score, precision, mAP@0.5, and recall rate.

4.2.1. Results of WIoU

Figures 11–13 illustrate the training outcomes of the YOLOv8 model optimized with the WIoU loss function. In a comparative analysis with the CIoU loss function, the WIoU-optimized model exhibits clear improvements across multiple performance metrics. Notably, the F_1-score increased from 0.84 to 0.85, marking a 1.2% enhancement, which suggests a better harmonic balance between precision and recall. While the precision at the optimal confidence level remained at 1.00 for both models, the WIoU model achieved this at a lower confidence threshold of 0.898, compared to 0.939 for the CIoU model. This lower threshold indicates enhanced model reliability at lower confidence levels, reducing the risk of missing true positives in practical applications.

Additionally, the mAP@0.5 saw a significant improvement from 0.876 to 0.895, representing a 2.2% increase, underscoring the model's superior ability to accurately detect cracks across various scales. These gains reflect the WIoU optimization's impact in refining the model's capacity to handle complex detection tasks by effectively weighting the intersection over union, leading to more stable and reliable results. Overall, the enhanced performance metrics of the WIoU-optimized YOLOv8 model, particularly in F_1-score and precision at varied confidence levels, confirm its effectiveness in improving both robustness and accuracy for object detection tasks in challenging environments.

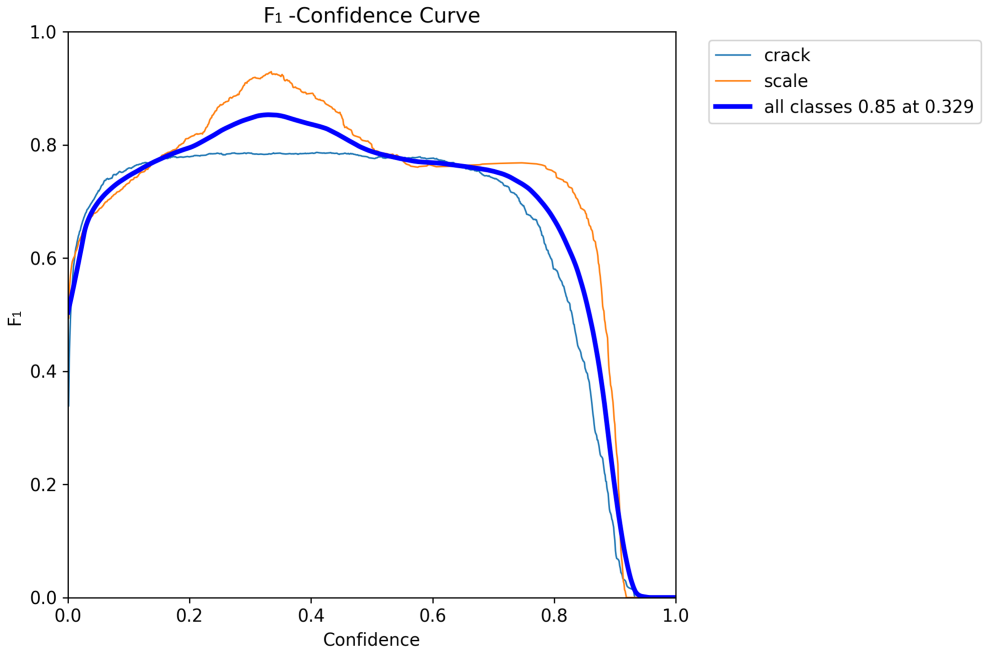

Figure 11. F_1-confidence curve (with WIoU applied).

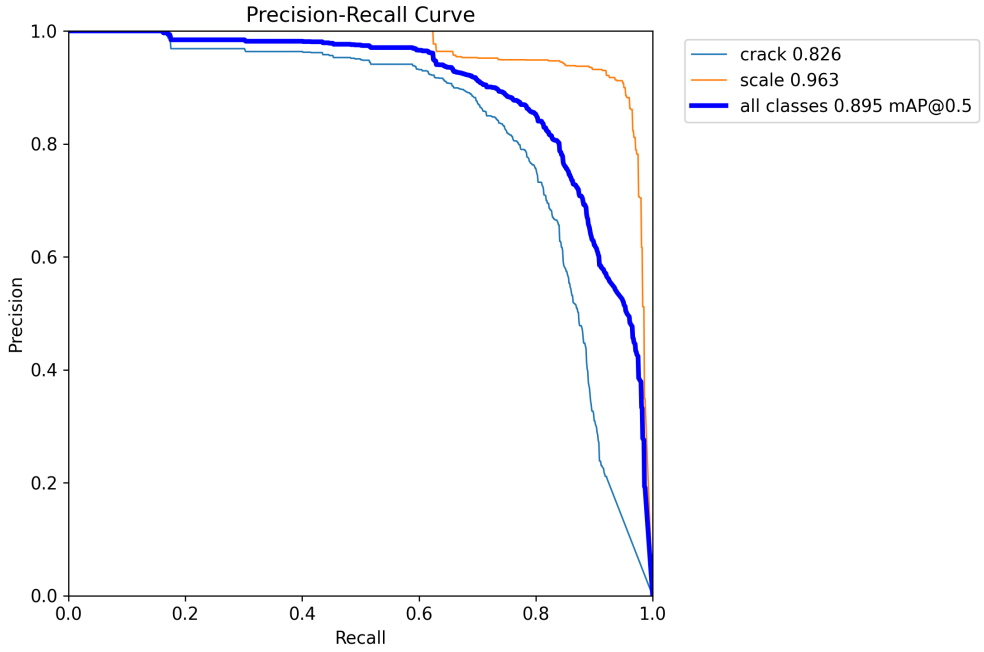

Figure 12. Precision–recall curve (with WIoU applied).

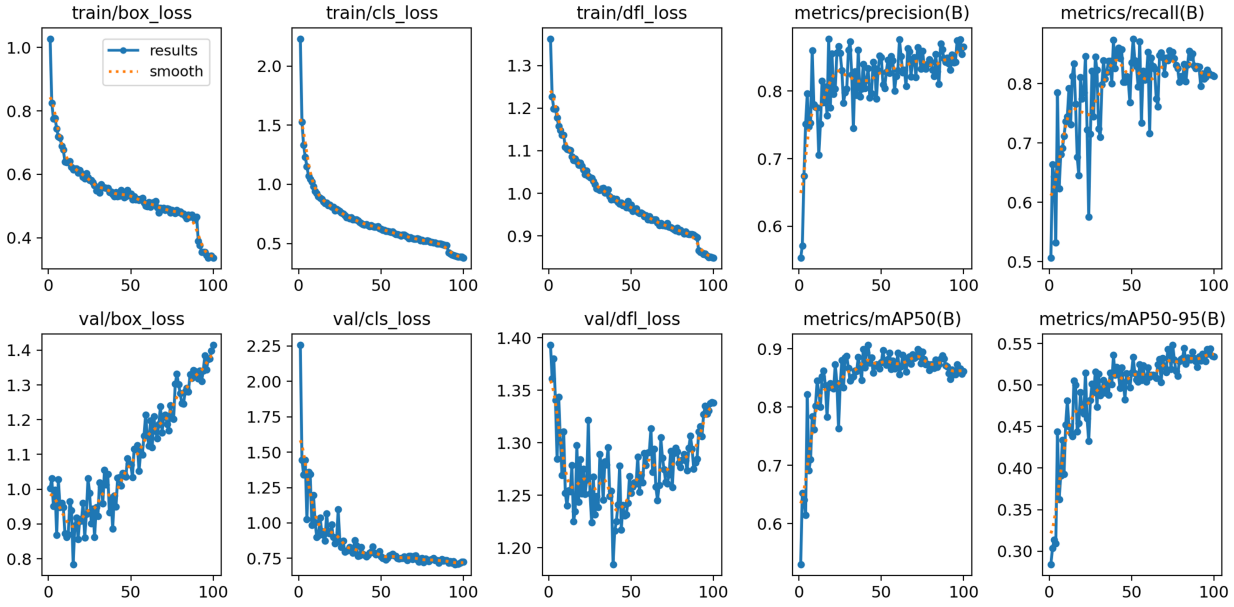

Figure 13. Comprehensive analysis of training and validation loss curves (left 3 columns), including precision, recall, and mean average precision, at various thresholds for model evaluation (right 2 columns), with WIoU applied.

Figure 14 shows the prediction results of improving CIoU to WIoU. After the improvement, the cracks that were originally undetectable, such as overlapping or unclear features, are detected. The results suggest that the WIoU optimization not only improves the overall detection performance but also enhances the model's sensitivity to subtle and intricate crack features, thereby reducing the incidence of false negatives and improving the overall reliability of the detection system. This advancement demonstrates a significant step forward in the field of crack detection, offering practical benefits for maintenance and safety assessments in various industries.

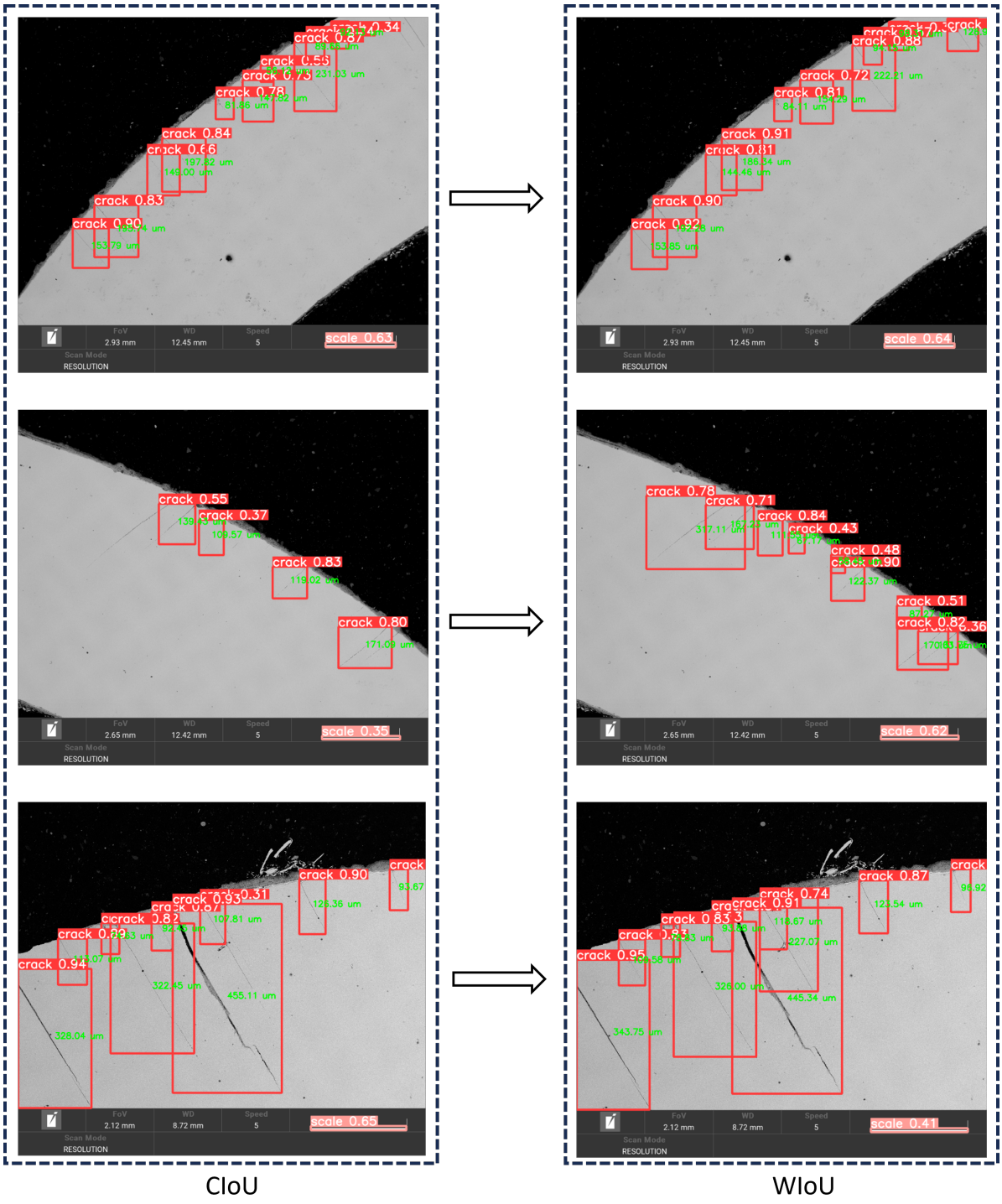

Figure 14. Detection performance comparison of CIoU vs. WIoU across crack scenarios.

4.2.2. Results of BiFPN

As shown in Figures 15–17, the YOLOv8 model has significantly improved in multiple performance indicators after the introduction of BiFPN. The best F_1-scores of both models are similar, both are 0.84, but BiFPN achieves this score at a slightly higher confidence threshold (0.386), and the F_1–confidence curve is more stable. The accuracy of both models is 1.00, but BiFPN achieves this accuracy at a lower confidence threshold (0.889), indicating that reliability has improved at lower confidence levels. The mAP@0.5 of BiFPN is 0.894, which is 2% higher than the original, and BiFPN shows better smoothness and consistency in the precision–recall curve. In addition, the training and validation loss curves of BiFPN

show a more stable downward trend, indicating that the loss is better controlled. The reduction of box loss is particularly obvious, indicating that the learning efficiency has improved. Overall, BiFPN improves the performance of the YOLOv8 model in various indicators, especially in terms of recall rate and loss stability, proving its effectiveness in improving detection capabilities.

The comparison results are shown in Figure 18, which shows that the optimized model is significantly better than the original model in terms of defect detection. Previously undetected defects are now accurately identified with the optimized model. Additionally, the original model exhibited incomplete bounding boxes for crack detection, leading to inaccuracies in length estimation and judgment. The optimized model effectively addresses these issues, providing more precise and complete crack detection, thereby improving the accuracy of defect characterization and measurement. This enhancement underscores the efficacy of the optimization in addressing the limitations of the original model and significantly advancing the reliability of defect detection.

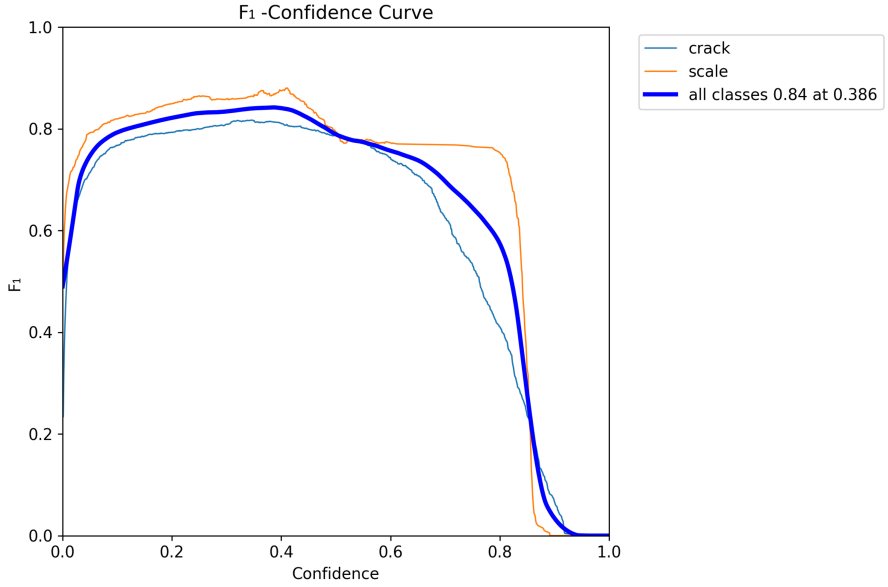

Figure 15. Confidence curve (with BiFPN applied).

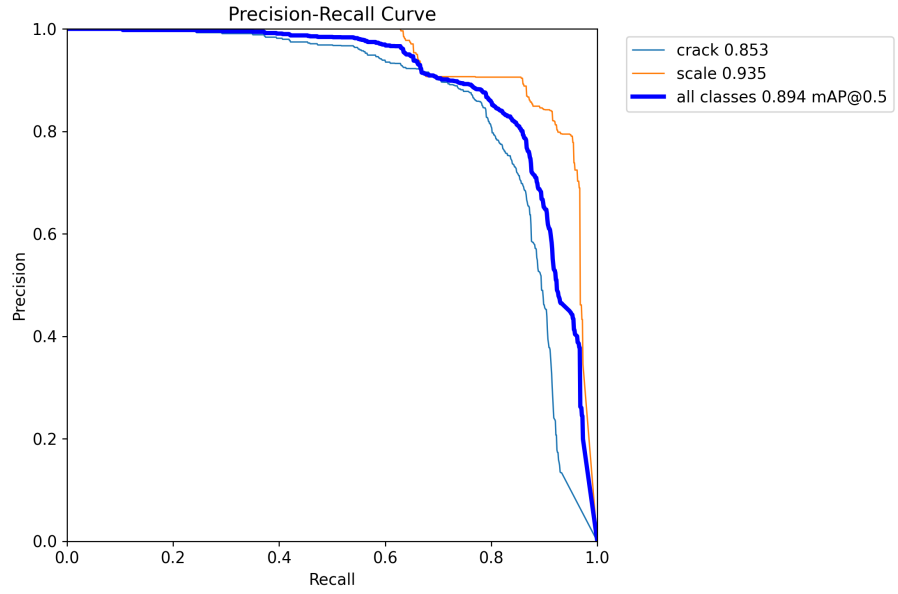

Figure 16. Precision–recall curve (with BiFPN applied).

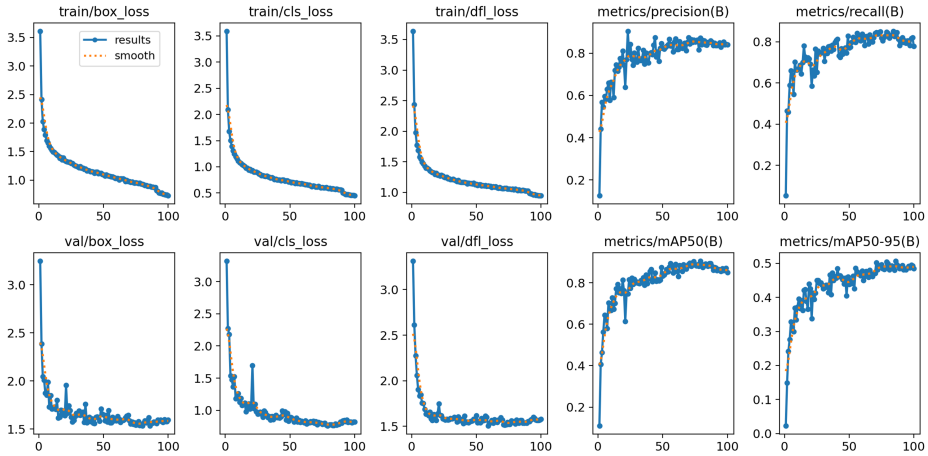

Figure 17. Comprehensive analysis of training and validation loss curves (left 3 columns), including precision, recall, and mean average precision, at various thresholds for model evaluation (right 2 columns), with BiFPN applied.

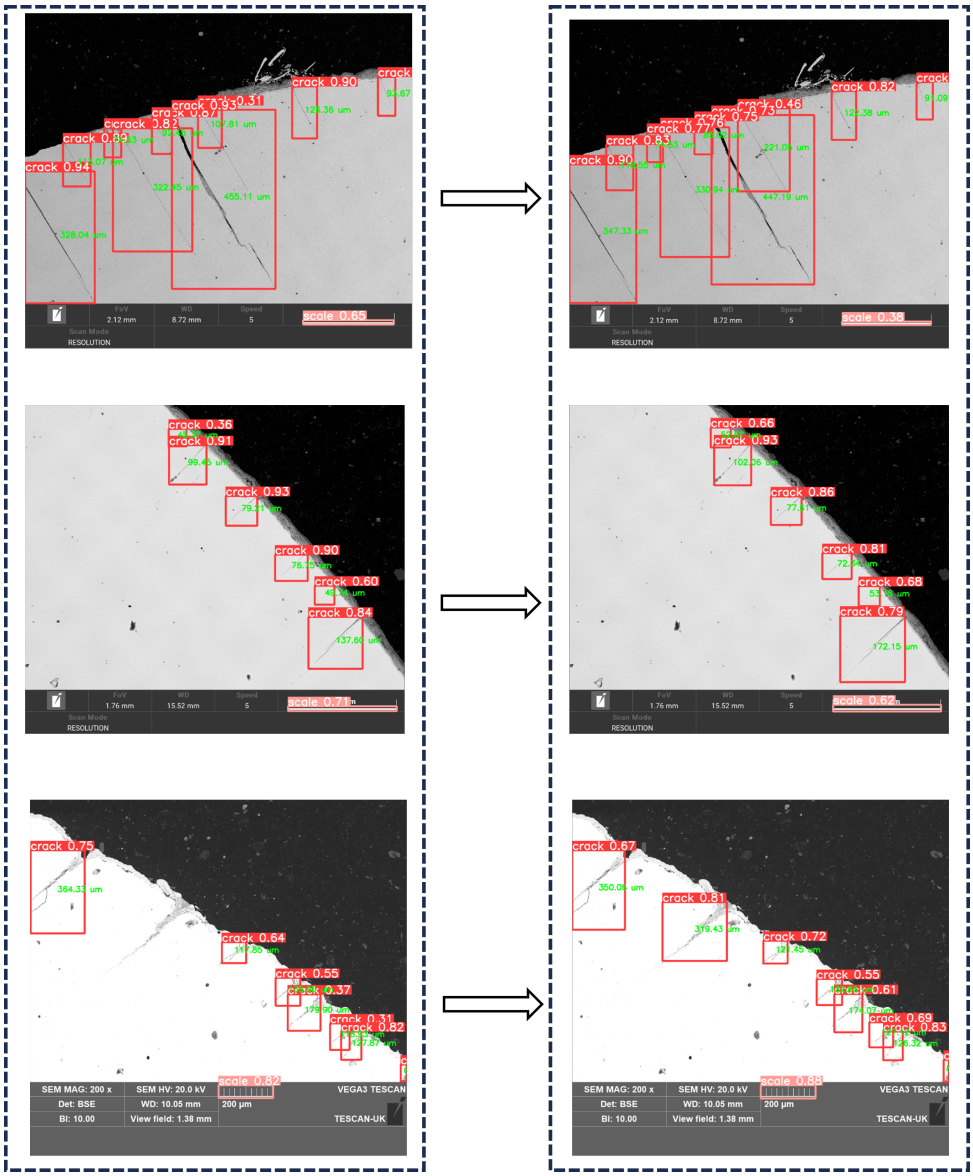

Figure 18. Detection results comparison: original method vs. BiFPN-enhanced model.

5. Discussion and Conclusions

Reflecting on the comparison between the results from the original and modified applications of the trained YOLOv8 model, we demonstrated that with the integration of WIoU and BiFPN, significant improvements in crack detection within SEM images were achieved. Figure 19 integrates the F_1 curves of the three methods, from which it can be seen that the introduction of the WIoU loss function significantly improves model performance, with the F_1-score increasing by 1.2%. Although the precision remains high at 1.00, it is achieved at a lower confidence threshold of 0.898 compared to 0.939 for the CIoU model, indicating improved reliability at lower confidence levels. Figure 20 integrates the PR curves of the three methods, with an increase of 2.2% in mAP@0.5, further demonstrating the positive impact of WIoU on the detection accuracy. The integration of BiFPN also helps improve performance. Although both models achieve similar best F_1-scores, the BiFPN optimized model achieves a mAP@0.5 of 0.894, which is 2% higher than the original model. As seen in Figure 15, the training and validation loss curves of the BiFPN-optimized model exhibit higher stability and reduced bounding box loss, reflecting enhanced learning efficiency.

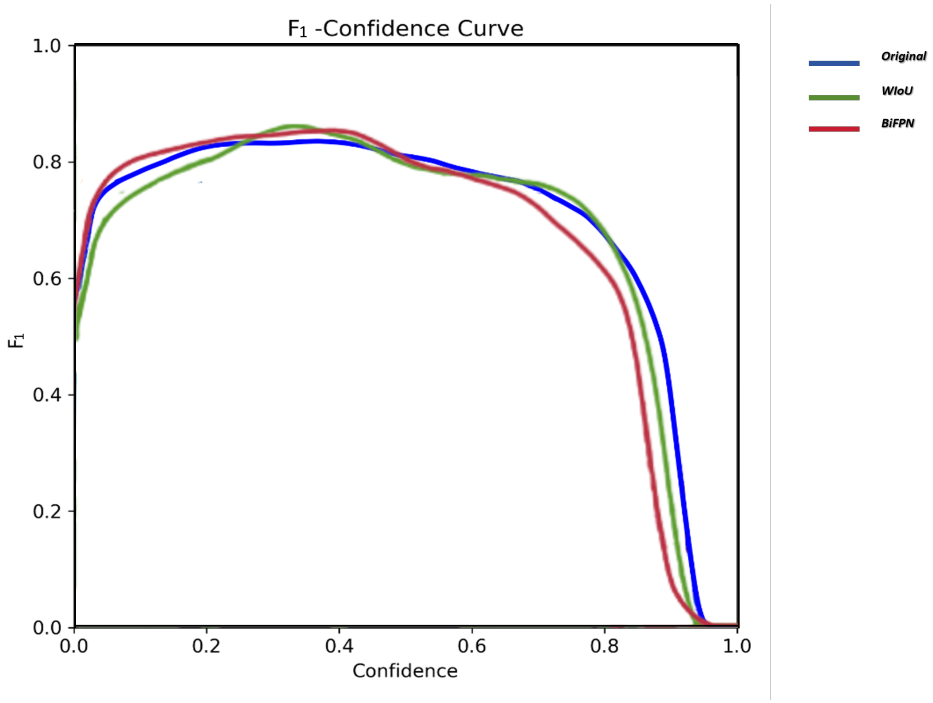

Figure 19. F_1 Curve comparison between original YOLOv8 output, and with the WIoU loss function and BiFPN technique outputs.

This study also explored other potential model improvements, such as incorporating omnidirectional convolution (ODConv, a convolution method that dynamically adjusts kernel weights, improving the model's ability to capture multi-scale and multi-directional features) [40], in the C2f module to replace traditional convolutions. Although the inclusion of ODConv resulted in improved recall rates and continued strong performance in label detection, the F_1-score did not show a substantial increase. Specifically, the mAP@0.5 improved by 1.6%, suggesting only a modest benefit. Additionally, the introduction of the Efficient Multi-Scale Context Attention (EMCA, an improved attention mechanism that adaptively adjusts the feature weights across different channels, enhancing the model's ability to extract features in complex backgrounds) [41]. The mechanism in the SPPF module resulted in an F_1-score increase of 1.2% and a 1.3% improvement in the mAP@0.5. However, the overall performance of this approach was still inferior to the WIoU and BiFPN methods. This may be attributed to the EMCA mechanism's tendency to optimize high

confidence areas, which does not perform as effectively when dealing with low confidence or complex backgrounds.

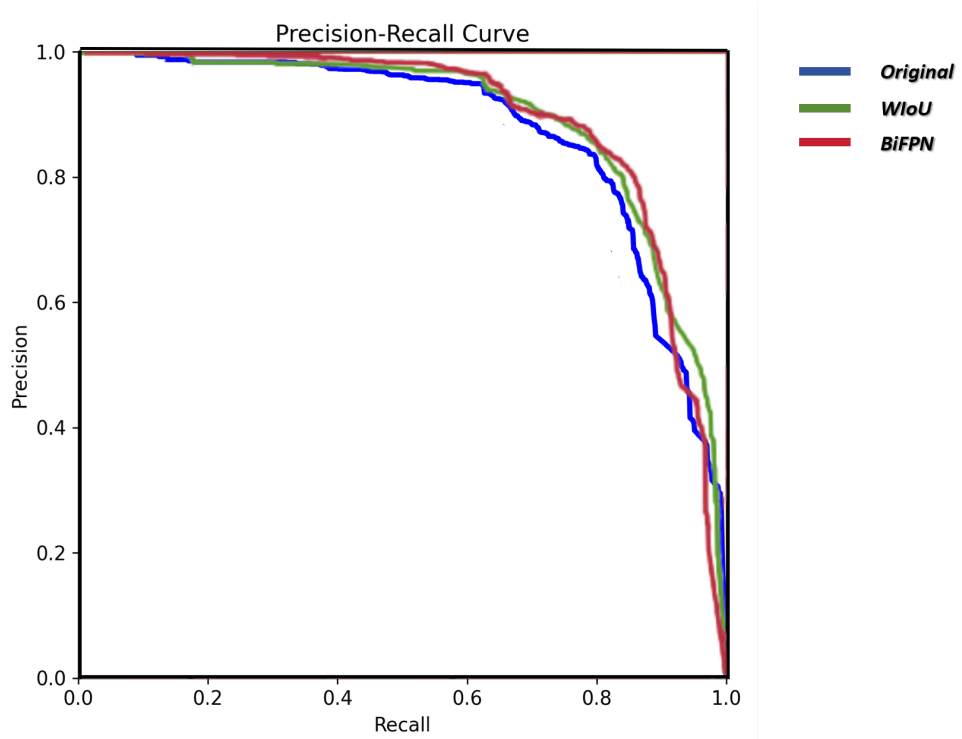

Figure 20. PR curve comparison between original YOLOv8 output, and with WIoU loss function and BiFPN technique outputs.

Based on these findings, this study further examined the combination of different optimization methods to enhance model performance. The comparison of model result parameters is shown in Table 3. Combining WIoU with ODConv showed a moderate improvement in precision but did not significantly enhance recall, resulting in intermediate performance compared to each method used individually. This suggests that the benefits of combining these two methods were not synergistic, as the increase in precision was counterbalanced by a decrease in recall. In contrast, combining BiFPN with ODConv led to decreased accuracy, likely due to conflicting optimization goals between BiFPN's multi-scale feature fusion and ODConv's adaptive kernel weighting. This conflict resulted in redundancy and hindered effective learning, ultimately degrading model performance. The increased complexity of these combinations also posed challenges, such as overfitting and higher training difficulty, which further reduced their effectiveness.

Table 3. Comparison of optimization results using 9 models.

Model	Maximum F_1-Score	Precision	Recall	mAP50	mAP50-95
Original	0.84	0.841	0.831	0.876	0.534
WIoU	0.85	0.859	0.854	0.895	0.548
BiFPN	0.84	0.852	0.839	0.894	0.506
ODConv	0.84	0.845	0.833	0.890	0.522
EMCA	0.85	0.842	0.854	0.888	0.544
WIoU-ODConv	0.84	0.856	0.832	0.887	0.532
BiFPN-ODConv	0.84	0.855	0.832	0.870	0.496
BiFPN-WIoU	0.83	0.845	0.830	0.868	0.502
EMCA-ODConv	0.80	0.858	0.756	0.849	0.511

Through optimization algorithms such as WIoU and BiFPN, the model's accuracy in SEM image crack detection was significantly enhanced, as reflected by improvements in the F_1-score, precision, mAP@0.5, and recall rate. The analysis in Table 3 highlights that WIoU and BiFPN individually provide the most notable improvements, with WIoU achieving the highest mAP@0.5, precision and recall. Moreover, as mentioned earlier, BiFPN also performs well in these aspects and has strong advantages in stability and reducing training loss. However, attempts to combine multiple optimization methods revealed that excessive combinations could lead to performance degradation. This suggests that while individual optimization strategies can effectively enhance model performance, combining different methods may result in conflicts between optimization strategies, leading to suboptimal learning and reduced accuracy. Therefore, it is crucial to exercise caution when selecting and combining optimization strategies to avoid potential negative effects.

Despite the considerable progress achieved, some limitations remain that warrant further investigation. Although the optimized model is more accurate, it still faces challenges in detecting very small or highly overlapping cracks. For instance, as shown in Figure 21, when crack features are unclear or overlap, prediction accuracy decreases. Additionally, while these optimizations have improved model performance, there is still room for enhancement in handling more complex and diverse backgrounds. Future research should focus on addressing these challenges, refining the optimization techniques, and conducting large-scale experiments. This approach would improve the model's versatility and robustness, making it more effective for defect detection across a variety of real-world applications.

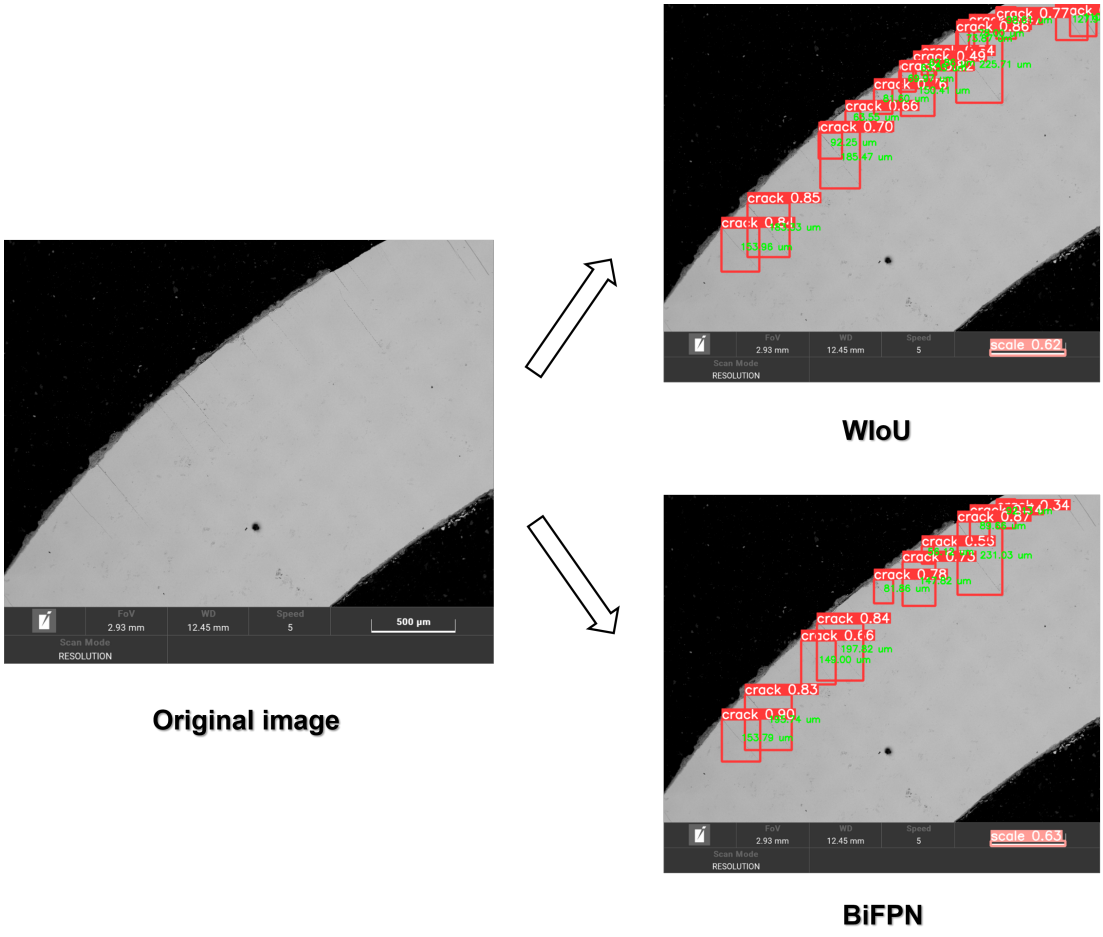

Figure 21. Enhanced model results with noted limitations in crack detection.

This study optimized the YOLOv8 model for crack detection in SEM images, integrating advanced techniques such as WIoU and BiFPN to enhance detection accuracy, speed, and model efficiency. The results demonstrate that these modifications significantly improve the model's capability to rapidly identify cracks, particularly in complex backgrounds, while maintaining precise length measurement. The findings of this study provide a solid foundation for future research in automated crack detection technologies. With ongoing refinement and expansion, the optimized YOLOv8 model can be further adapted and applied across various industrial inspection tasks, promoting rapid safety assessments and reliability in various fields.

Further Work

Although this study has made significant progress in crack detection by optimizing the YOLOv8 model, several areas remain open for further exploration. First, future research could focus on enhancing the model's robustness in handling complex backgrounds and low-contrast cracks. While the optimized model performs well in most scenarios, its accuracy still declines when detecting very small or highly overlapping cracks. To address these challenges, further adjustments to the model's architecture or the adoption of advanced feature extraction techniques, such as multi-scale feature fusion or self-attention mechanisms, could be explored.

Improving the model's real-time processing capability and adaptability should also be a priority for future research. In industrial applications, models must offer not only high accuracy but also fast inference speed and low latency. Therefore, optimizing the model's computational efficiency without sacrificing accuracy is crucial. Another key area for future work is investigating the model's ability across different scenarios, such as its performance in crack detection across various materials and environmental conditions.

Finally, future studies could explore applying this model to a broader range of fields, including road crack detection, building structural health monitoring, lesion detection in medical imaging, and crack detection in railways, aerospace, and other industrial machinery. Conducting experiments in these areas would help validate the model's versatility and adaptability, expanding its scope of application and providing valuable technical support for research in diverse fields.

Author Contributions: Conceptualization, C.L., L.T. and B.H.S.A.; methodology, C.L., L.T. and B.H.S.A.; resources, C.L., F.D.M. and S.G.; writing—original draft preparation, C.L.; writing—review and editing, L.T., B.H.S.A. and C.L.; supervision, L.T. and B.H.S.A.; visualization, L.T. and B.H.S.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Data are not available due to privacy or ethical restrictions.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this paper:

CNN	Convolutional neural network
SEM	Scanning electron microscopy
YOLO	You Only Look Once
WIoU	Wise Intersection over Union
FPN	Feature pyramid network
BiFPN	Bidirectional feature pyramid network
CSP	Cross Stage Partial
SPPF	Spatial Pyramid Pooling Fast
PANet	Path Aggregation Network
VFL	Varifocal Loss
DFL	Distance-Focal L1

CBAM	Convolutional Block Attention Module
SE	Squeeze-and-Excitation
SA	Self-Attention
IoU	Intersection over Union
CIoU	Complete Intersection over Union
DIoU	Distance Intersection over Union
GIoU	Generalized Intersection over Union
SIoU	Shape-Aware Intersection over Union
RepVGG	Re-parameterizable VGG Network
BBR	Bounding Box Regression
CIoU	Complete Intersection over Union
NAS	Neural Architecture Search
ODConv	Omnidirectional Convolution
EMCA	Efficient Multi-Scale Context Attention
SPPF	Spatial Pyramid Pooling - Fast
CLAHE	Contrast Limited Adaptive Histogram Equalization
mAP	Mean Average Precision
box_loss	Bounding Box Loss
cls_loss	Classification Loss
dfl_loss	Distribution Focal Loss

References

1. Mohan, A.; Poobal, S. Crack detection using image processing: A critical review and analysis. *Alex. Eng. J.* **2018**, *57*, 787–798. [CrossRef]
2. Wu, X.; Sahoo, D.; Hoi, S.C. Recent advances in deep learning for object detection. *Neurocomputing* **2020**, *396*, 39–64. [CrossRef]
3. Deng, J.; Singh, A.; Zhou, Y.; Lu, Y.; Lee, V.C.S. Review on computer vision-based crack detection and quantification methodologies for civil structures. *Constr. Build. Mater.* **2022**, *356*, 129238. [CrossRef]
4. Xu, G.; Yue, Q.; Liu, X. Deep learning algorithm for real-time automatic crack detection, segmentation, qualification. *Eng. Appl. Artif. Intell.* **2023**, *126*, 107085. [CrossRef]
5. Jiang, Y.; Zhao, C. Attention classification-and-segmentation network for micro-crack anomaly detection of photovoltaic module cells. *Sol. Energy* **2022**, *238*, 291–304. [CrossRef]
6. Melanthota, S.K.; Gopal, D.; Chakrabarti, S.; Kashyap, A.A.; Radhakrishnan, R.; Mazumder, N. Deep learning-based image processing in optical microscopy. *Biophys. Rev.* **2022**, *14*, 463–481. [CrossRef]
7. Liu, X.; Song, L.; Liu, S.; Zhang, Y. A review of deep-learning-based medical image segmentation methods. *Sustainability* **2021**, *13*, 1224. [CrossRef]
8. Talaei Khoei, T.; Ould Slimane, H.; Kaabouch, N. Deep learning: Systematic review, models, challenges, and research directions. *Neural Comput. Appl.* **2023**, *35*, 23103–23124. [CrossRef]
9. Hu, X.; Chu, L.; Pei, J.; Liu, W.; Bian, J. Model complexity of deep learning: A survey. *Knowl. Inf. Syst.* **2021**, *63*, 2585–2619. [CrossRef]
10. Pauly, L.; Hogg, D.; Fuentes, R.; Peel, H. Deeper networks for pavement crack detection. In Proceedings of the 34th ISARC, IAARC, Taipei, Taiwan, 28 June–1 July 2017; pp. 479–485.
11. Li, H.; Zhang, H.; Zhu, H.; Gao, K.; Liang, H.; Yang, J. Automatic crack detection on concrete and asphalt surfaces using semantic segmentation network with hierarchical Transformer. *Eng. Struct.* **2024**, *307*, 117903. [CrossRef]
12. Zhang, P.; Ryu, H.; Miao, Y.; Jo, S.; Park, G. Robust unsupervised-learning based crack detection for stamped metal products. *J. Manuf. Syst.* **2024**, *73*, 65–74. [CrossRef]
13. Kou, X.; Liu, S.; Cheng, K.; Qian, Y. Development of a YOLO-V3-based model for detecting defects on steel strip surface. *Measurement* **2021**, *182*, 109454. [CrossRef]
14. Hu, H.; Li, Z.; He, Z.; Wang, L.; Cao, S.; Du, W. Road surface crack detection method based on improved YOLOv5 and vehicle-mounted images. *Measurement* **2024**, *229*, 114443. [CrossRef]
15. Su, P.; Han, H.; Liu, M.; Yang, T.; Liu, S. MOD-YOLO: Rethinking the YOLO architecture at the level of feature information and applying it to crack detection. *Expert Syst. Appl.* **2024**, *237*, 121346. [CrossRef]
16. Swathi, Y.; Challa, M. YOLOv8: Advancements and Innovations in Object Detection. In Proceedings of the International Conference on Smart Computing and Communication, Denpasar, Indonesia, 25–27 July 2024; Springer: Berlin/Heidelberg, Germany, 2024; pp. 1–13.
17. Terven, J.; Córdova-Esparza, D.M.; Romero-González, J.A. A comprehensive review of yolo architectures in computer vision: From yolov1 to yolov8 and yolo-nas. *Mach. Learn. Knowl. Extr.* **2023**, *5*, 1680–1716. [CrossRef]
18. Cao, X.; Duan, M.; Ding, H.; Yang, Z. MS-YOLO: Integration-based multi-subnets neural network for object detection in aerial images. *Earth Sci. Inform.* **2024**, *17*, 2085–2106. [CrossRef]

19. Zhu, J.; Hu, T.; Zheng, L.; Zhou, N.; Ge, H.; Hong, Z. YOLOv8-C2f-Faster-EMA: An Improved Underwater Trash Detection Model Based on YOLOv8. *Sensors* **2024**, *24*, 2483. [CrossRef]
20. Sohan, M.; Sai Ram, T.; Reddy, R.; Venkata, C. A review on yolov8 and its advancements. In Proceedings of the International Conference on Data Intelligence and Cognitive Informatics, Tirunelveli, India, 18–20 November 2024; Springer: Berlin/Heidelberg, Germany, 2024; pp. 529–545.
21. Hussain, M. YOLOv5, YOLOv8 and YOLOv10: The Go-To Detectors for Real-time Vision. *arXiv* **2024**, arXiv:2407.02988 .
22. Yaseen, M. What is YOLOv8: An In-Depth Exploration of the Internal Features of the Next-Generation Object Detector. *arXiv* **2024**, arXiv:2408.15857.
23. Yu, S.; Fan, H.; Yang, Y.; Tang, Y.; Sun, R.; Xu, W. Indoor scene recognition based on YOLOv8 improved loss function. In Proceedings of the 2023 Asia Conference on Advanced Robotics, Automation, and Control Engineering (ARACE), Chengdu, China, 18–20 August 2023; IEEE: Piscataway, NJ, USA, 2023; pp. 55–61.
24. Liu, Z.; Ye, K. YOLO-IMF: An improved YOLOv8 algorithm for surface defect detection in industrial manufacturing field. In Proceedings of the International Conference on Metaverse, Honolulu, HI, USA, 23–26 September 2023; Springer: Berlin/Heidelberg, Germany, 2023; pp. 15–28.
25. Wang, G.; Chen, Y.; An, P.; Hong, H.; Hu, J.; Huang, T. UAV-YOLOv8: A small-object-detection model based on improved YOLOv8 for UAV aerial photography scenarios. *Sensors* **2023**, *23*, 7190. [CrossRef]
26. Woo, S.; Park, J.; Lee, J.Y.; Kweon, I.S. Cbam: Convolutional block attention module. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 3–19.
27. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7132–7141.
28. Zhang, Q.L.; Yang, Y.B. Sa-net: Shuffle attention for deep convolutional neural networks. In Proceedings of the ICASSP 2021–2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Toronto, ON, Canada, 6–11 June 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 2235–2239.
29. Zheng, Z.; Wang, P.; Liu, W.; Li, J.; Ye, R.; Ren, D. Distance-IoU loss: Faster and better learning for bounding box regression. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 12993–13000.
30. Rezatofighi, H.; Tsoi, N.; Gwak, J.; Sadeghian, A.; Reid, I.; Savarese, S. Generalized intersection over union: A metric and a loss for bounding box regression. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 658–666.
31. Gevorgyan, Z. SIoU loss: More powerful learning for bounding box regression. *arXiv* **2022**, arXiv:2205.12740.
32. Ding, X.; Zhang, X.; Ma, N.; Han, J.; Ding, G.; Sun, J. Repvgg: Making vgg-style convnets great again. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 13733–13742.
33. Yan, J.; Wang, H.; Yan, M.; Diao, W.; Sun, X.; Li, H. IoU-adaptive deformable R-CNN: Make full use of IoU for multi-class object detection in remote sensing imagery. *Remote Sens.* **2019**, *11*, 286. [CrossRef]
34. Rosebrock, A. Intersection over Union (IoU) for Object Detection. 2016. Available online: https://www.pyimagesearch.com/2016/11/07/intersection-over-union-iou-for-object-detection/ (accessed on 18 May 2021).
35. Zheng, Z.; Wang, P.; Ren, D.; Liu, W.; Ye, R.; Hu, Q.; Zuo, W. Enhancing geometric factors in model learning and inference for object detection and instance segmentation. *IEEE Trans. Cybern.* **2021**, *52*, 8574–8586. [CrossRef] [PubMed]
36. Du, S.; Zhang, B.; Zhang, P.; Xiang, P. An improved bounding box regression loss function based on CIOU loss for multi-scale object detection. In Proceedings of the 2021 IEEE 2nd International Conference on Pattern Recognition and Machine Learning (PRML), Chengdu, China, 16–18 July 2021; IEEE: Piscataway, NJ, USA , 2021; pp. 92–98.
37. Gao, S.; Chu, M.; Zhang, L. A detection network for small defects of steel surface based on YOLOv7. *Digit. Signal Process.* **2024**, *149*, 104484. [CrossRef]
38. Tong, Z.; Chen, Y.; Xu, Z.; Yu, R. Wise-IoU: Bounding box regression loss with dynamic focusing mechanism. *arXiv* **2023**, arXiv:2301.10051.
39. Tan, M.; Pang, R.; Le, Q.V. Efficientdet: Scalable and efficient object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 10781–10790.
40. Li, C.; Zhou, A.; Yao, A. Omni-dimensional dynamic convolution. *arXiv* **2022**, arXiv:2209.07947.
41. Bakr, E.M.; El-Sallab, A.; Rashwan, M. EMCA: Efficient multiscale channel attention module. *IEEE Access* **2022**, *10*, 103447–103461. [CrossRef]

Optimized AI methods for rapid crack detection in microscopy images

Lou, Chenxukun

2024-12-06

Lou C, Tinsley L, Duarte Martinez F, et al., (2024) Optimized AI methods for rapid crack detection in microscopy images. Electronics, Volume 13, Issue 23, Article number 4824 https://doi.org/10.3390/electronics13234824