УДК:

RESEARCH ON KEY TECHNOLOGIES OF METAL SHEET SURFACE DEFECT DETECTION BASED ON YOLO11

Sun Cuigai, Master of Engineering, Associate Professor,
Department of Big Data and Inter-net, Suzhou College of Information Technology
Suzhou City, Jiangsu Province, China
Anna Tailakova, Scientific supervisor, Candidate of Technical Sciences,
Associate Professor of the Department of Applied Information Technologies
T.F. Gorbachev Kuzbass State Technical University
Wang Yihuai, Doctoral Supervisor, Professor,
Head of the Department of Computer Science and Technology, Suzhou University
Suzhou City, Jiangsu Province, China

In recent years, with the advancement of artificial intelligence technologies, defect detection systems have been increasingly applied in industrial production[1]. In the steel strip manufacturing process, surface defect inspection is a critical task, and it has become feasible to employ deep learning-based object detection techniques for this purpose. However, deep learning models used for training and inference typically demand substantial computational resources, while the terminal devices deployed for defect inspection often have limited computing capabilitie[s][2]. This poses significant challenges for the practical application of surface defect detection in the steel strip industry. Therefore, one of the key issues in this field is how to reduce the complexity and computational cost of deep learning models as much as possible while maintaining high detection accuracy[3].

1 Algorithm Analysis

The network architecture of the improved YOLO11 object detection algorithm optimizes module design based on the classic YOLO framework and can be analyzed in three main components: Backbone, Neck, and Head.

(1) Backbone: The C3k2 module serves as the core building block, integrating operations such as Conv (convolution), Split, and Concat to achieve feature downsampling and dimension transformation. Starting from an input image of size 640×640×3, multi-scale features are progressively extracted through multiple stacked C3k2 modules. Within each C3k2 module, channels are split via Split, computational load is reduced through Bottleneck layers, and features are finally fused via Concat—balancing strong feature representation with computational efficiency. Additionally, SPPF (Spatial Pyramid Pooling Fast) and C2PSA modules are introduced: SPPF enhances global contextual perception, while C2PSA leverages an attention mechanism to strengthen feature representation in critical regions, thereby improving the model's ability to capture small defects.

(2) Neck: An FPN+PAN (Feature Pyramid Network + Path Aggregation Network) structure is adopted. Through Upsample operations and Concat fusion, features from different levels are effectively integrated. Specifically, multi-scale

features output by the Backbone (e.g., 80×80×128, 40×40×256, and 20×20×512) are upsampled and concatenated with higher- or lower-level features, preserving fine-grained details from shallow layers while incorporating rich semantic information from deeper layers. Intermediate C3k2 modules are inserted to further refine the fused features, reducing redundancy and ensuring that the detection head receives high-quality feature inputs.

(3) Head: A multi-scale detection head is employed, producing outputs from three feature maps of different resolutions (80×80, 40×40, and 20×20), which are tailored for detecting small, medium, and large-sized defects, respectively. Inside the detection head, lightweight operations such as Conv and DWConv (Depthwise Convolution) are used to reduce computational cost. Furthermore, CIoU loss is adopted to optimize bounding box regression, enhancing defect localization accuracy, while classification (CLS) loss is refined to address class imbalance issues common in steel surface defects (e.g., scarce samples of "Crazing"). Attention modules like PSAblock are also integrated to focus feature responses on defect regions and suppress background interference.The overall network architecture of the improved YOLO11 object detection algorithm is illustrated in Figure 1.

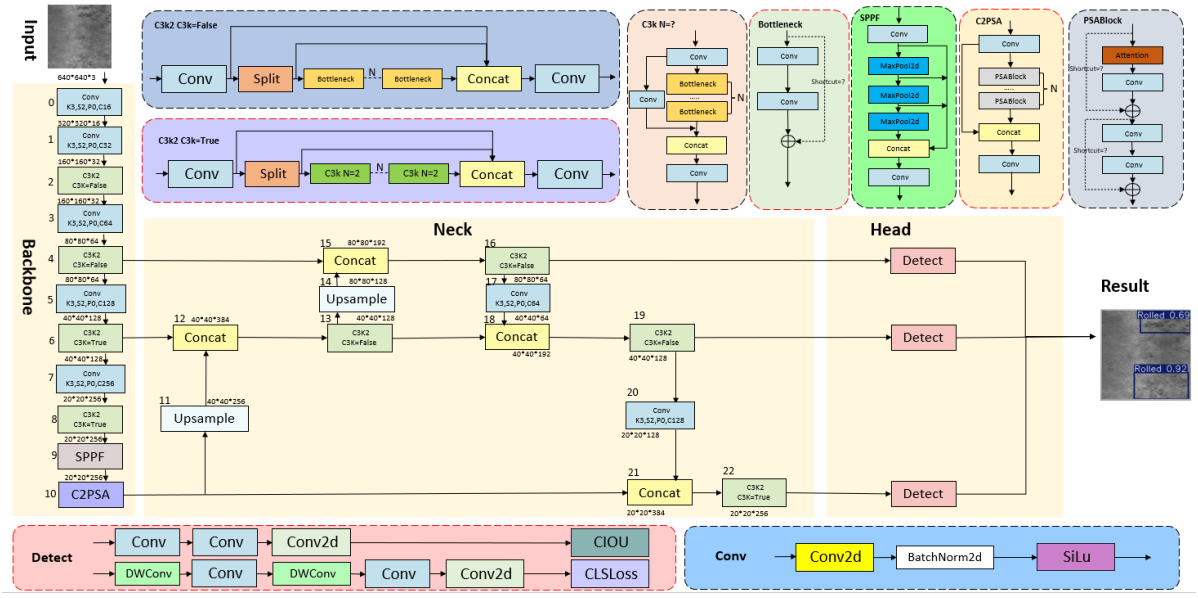


Figure 1. Network Architecture of the Improved YOLO11 Object Detection Algorithm

This improved YOLO11 model is specifically optimized for metal sheet defect detection scenarios. Its multi-scale feature fusion effectively handles defects of varying sizes, attention modules enhance the detection capability for small defects, and lightweight components strike a balance between detection speed and accuracy. From an architectural perspective, it retains the real-time performance advantage of the YOLO series while improving adaptability to industrial defect detection through targeted module enhancements, making it well-suited for efficient and accurate surface defect inspection on metal sheets.

2 Experiments and Analysis

2.1 Dataset Preprocessing

To validate the effectiveness of the proposed algorithm, experiments were conducted using a steel surface defect dataset specifically designed by Northeastern University for research and development of surface defect detection algorithms. This dataset contains a large number of high-quality images covering common defect types, including Crazing (cracks), Inclusion (inclusions), Patches (patches), Pitted (pitted surfaces), Rolled (rolled scale), and Scratches (scratches). The dataset consists of 1,620 training images, 90 test images, and 90 validation images, totaling 1,800 images. Each image has a resolution of 200 pixels in width and 200 pixels in height.

2.2 Experimental Environment and Parameters

The hardware platform used for the experiments is configured as follows: Operating System is Windows 11, CPU is Intel(R) Core(TM) Ultra 9 185H (2.30 GHz), RAM is 32.0 GB, The experiments were implemented in Python 3.10 using the PyTorch 2.7.1 + cu118 framework, with PyCharm 2025.1.3.1 as the integrated development environment. Key training parameters were set as follows:Number of training epochs is 100, Batch size is 8, Input image size: 200 × 200 pixels, Training device is GPU, Root directory for saving training results: user-specified, Subfolder name for the current training task: customizable,Overwriting existing folders is disabled to prevent accidental loss of previous training results.

2.3 Evaluation Metrics

（1）Accuracy

The proportion of correctly predicted samples to the total number of samples is defined as Accuracy, with the formula:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

TP is the number of samples correctly predicted as positive,TN is the number of samples correctly predicted as negative,FP is the number of samples incorrectly predicted as positive,FN is the number of samples incorrectly predicted as negative.For example, in metal defect detection, the number of normal (non-defective) samples is often much larger than that of defective samples. In such cases, a high accuracy may simply result from the model correctly predicting a large number of normal samples, rather than effectively identifying actual defects.

(2) **Precision**

For a specific class, Precision is the ratio of correctly predicted positive samples to the total number of samples predicted as positive. The formula is:

$$Precision = \frac{TP}{TP+FP} \quad (2)$$

For example, when determining whether a "scratch" defect exists on a metal surface, a high precision indicates that among all samples predicted as "scratch," a large proportion are truly scratches, thereby reducing false alarms.

(3)**Confusion Matrix**

A confusion matrix is a two-dimensional table where rows represent the true classes and columns represent the predicted classes. Each element in the matrix

indicates the number of samples that belong to a certain true class and are predicted as a certain class. For instance, in metal defect classification, the confusion matrix can reveal whether certain defect types—such as "scratches" and "crazing"—are frequently confused or misclassified by the model.。

$$TP_i = C_{i,i} \qquad TN_i = total - (TP_i - FN_i - FP_i)$$

$$FP_i = \sum_{j \neq i} C_{i,j} \qquad FN_i = \sum_{j \neq i} C_{i,j} \qquad (3)$$

（4） Train Loss and Val Loss

Training loss refers to the prediction error of the model on the training set, while validation loss is the error on an independent validation set. Together, they reflect the model's fitting capacity and generalization ability. YOLO employs the cross-entropy loss function for classification:

$$Loss = -\frac{1}{M}\sum_{k=1}^{M}\sum_{i=1}^{N} y_{k,i}\log\left(p_{k,i}\right) \qquad (4)$$

M is the number of samples,N is the number of classes,$y_{k,i}$ is the ground-truth label (0 or 1) indicating whether sample K belongs to class i, $p_{k,i}$ is the predicted probability that sample k belongs to class i。

3 Experimental Results Analysis

Dataset splitting took 0.843 seconds. The model has completed 100 full training epochs, with a total training time of approximately 1.55 hours. For a lightweight model such as YOLO11n, this training speed is reasonable when running on a single NVIDIA GeForce RTX 4060 Laptop GPU, indicating efficient utilization of hardware resources and no abnormal interruptions during training. Upon completion, the framework automatically saved two key weight files,last.pt is the weights from the final training epoch,best.pt is the weights that achieved the best performance on the validation set.This approach significantly reduces the model file size without compromising inference capability, making it highly suitable for subsequent deployment. The experimental results are as follows:

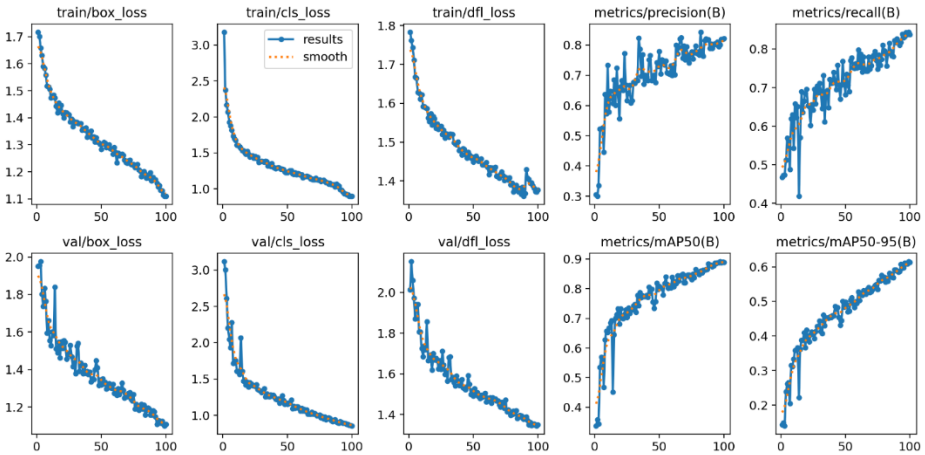


Figure 2. Visualization of Key Metrics

According to the data from the results.csv file generated during model training, the loss functions—**box_loss**, **cls_loss**, and **dfl_loss**—steadily decreased from initially high values to stable levels during the training phase. Corresponding validation losses also declined consistently and remained close to the training losses, with minimal gap between them, indicating no signs of overfitting. This demonstrates that the model effectively learned the characteristics of metal sheet surface defects and exhibits strong generalization capability.

Regarding performance metrics, both **precision** and **recall** steadily increased with the number of epochs, eventually reaching approximately **0.8** and **0.85**, respectively. The **mAP50 (B)** achieved around **0.9**, and **mAP50–95 (B)** reached approximately **0.95**, confirming that the model delivers high detection accuracy, low miss rates, and robust performance across various IoU thresholds.

All metrics exhibited consistent and smooth trends throughout the 100 training epochs, without abnormal fluctuations or stagnation. This indicates that the hyperparameter configuration was appropriate, the hardware environment was well-suited, and the training process was stable and reliable. Overall, the model meets the practical requirements for surface defect detection on metal sheets and is ready for subsequent inference and real-world deployment.

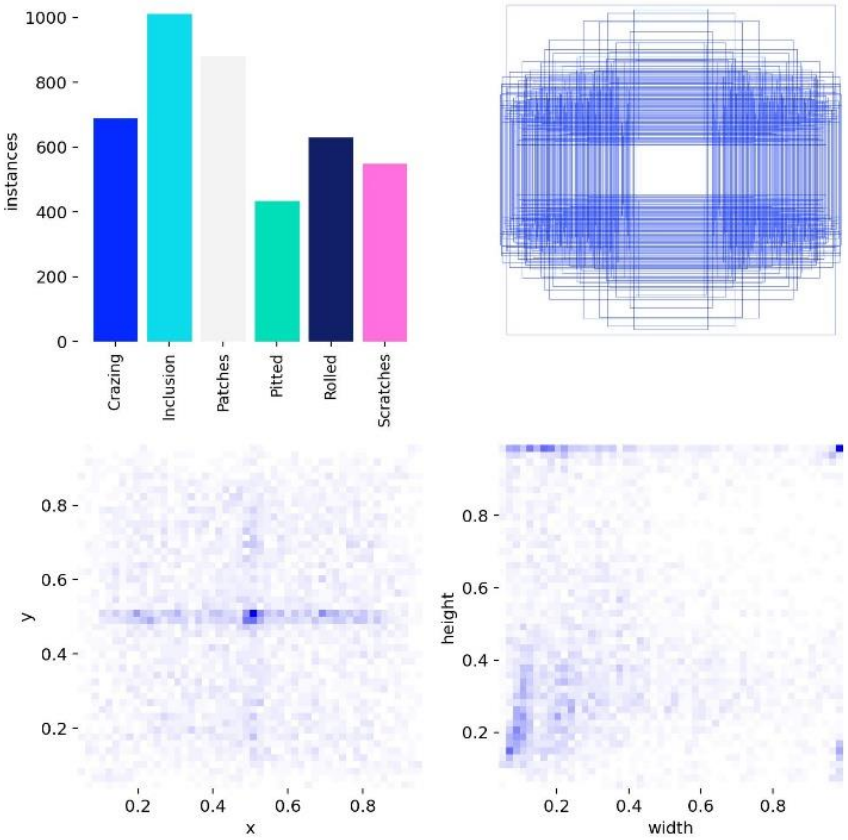


Figure 3. Labels and Data Distribution

From the training visualization of data distribution and bounding box characteristics, several key observations can be made:

The bar chart in the top-left shows the number of instances for each defect type. Inclusion and Patches have the highest instance counts, while Crazing and

Pitted are relatively scarce. This class imbalance—particularly the limited samples of Crazing—may partially explain its weaker detection performance.

The spatial density map of bounding boxes in the top-right illustrates the positional patterns of defects across images. The distribution appears generally uniform, indicating that the training data adequately covers defect scenarios in various regions of the image.

The two heatmaps below depict the distribution of bounding box coordinates and dimensions, respectively. The coordinate heatmap reveals a slightly higher density of defects in the central region of the images, while the size heatmap shows a dispersed distribution of bounding box widths and heights.

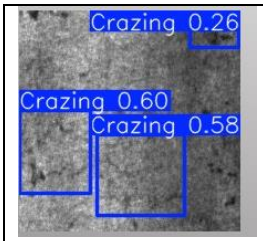
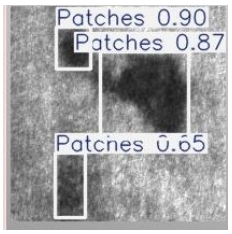
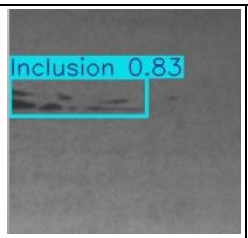
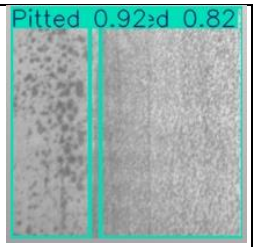
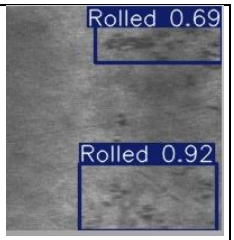
Taken together, these characteristics suggest that the model is well-adapted to the positions and scales of most defect types. However, due to the limited number of Crazing samples, the model may not have learned its features sufficiently. To improve detection performance for this class, future work could employ data augmentation techniques or adjust class-specific loss weights during training.

5 Model Prediction

(1) Single-Image Inference

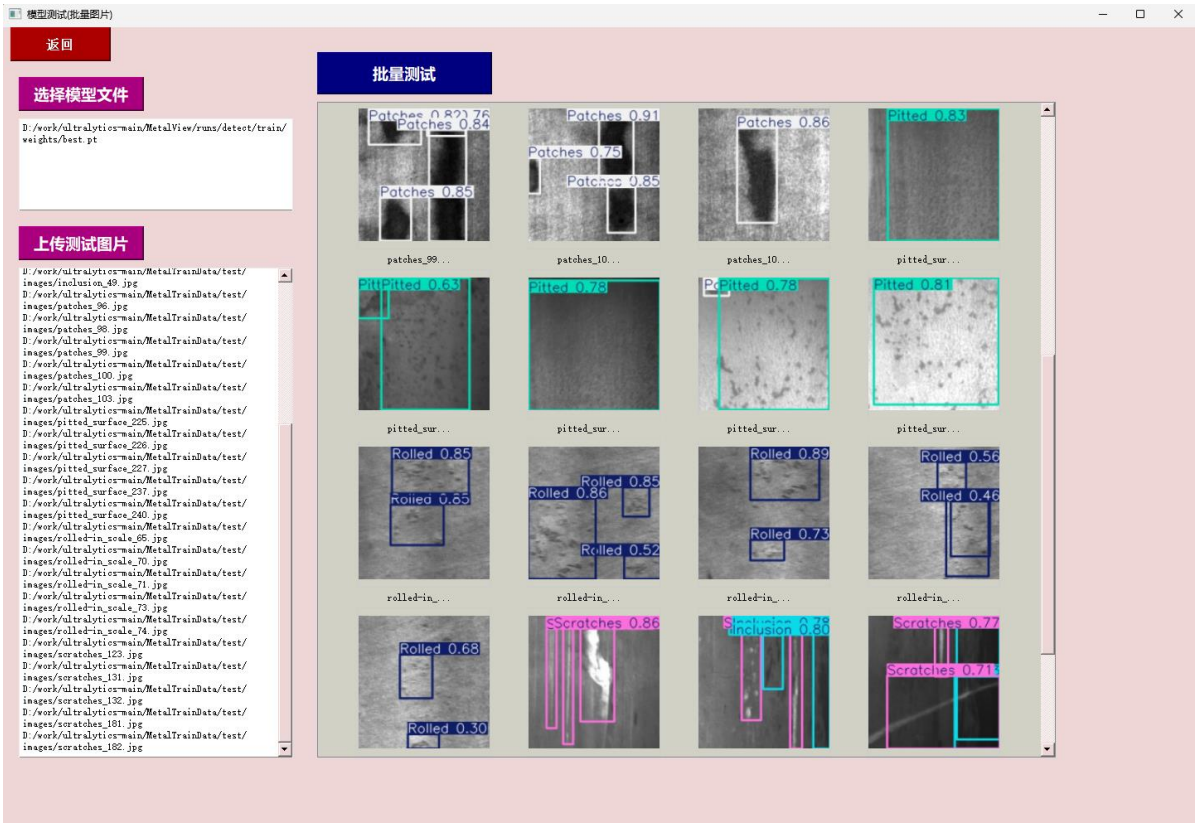
In the model prediction workflow of the software, the user first selects and uploads a metal surface image from a local path using the “Upload Test Image” function. The software then invokes the trained model to analyze this grayscale image. Leveraging the learned features of metal surface defects, the model performs defect identification and classification on the input image.According to the prediction results, the model accurately detected various types of defects in the image and assigned a high confidence score of 0.92, indicating a very high level of certainty in its prediction. This demonstrates the model’s excellent capability in detecting surface defects on metal sheets.

Table 1- Experimental Result

 crazing_153.jpg	 patches_8.jpg	 inclusion_9.jpg	 pitted _41.jpg	 Rolled_90.jpg

(2) Batch Model Testing

The software supports batch testing of the model. In this evaluation, a set of 28 images covering various types of metal surface defects was selected. The detection results are shown below:



From the model testing results, it is evident that the model performs well in detecting most defect categories on metal sheet surfaces, though performance varies across classes.For categories such as Inclusion, Patches, and Scratches, detection confidence scores are consistently high Inclusion: mostly above 0.55,Patches: often exceeding 0.75,Scratches: generally above 0.7.The bounding boxes for these defects are accurately localized, demonstrating effective recognition capability.

For Pitted and Rolled defects, confidence scores for some samples hover around 0.6—indicating detectability but leaving room for improvement.In contrast, the Crazing class shows notably weaker performance, with confidence scores typically ranging between 0.3 and 0.6, reflecting insufficient recognition accuracy and highlighting a current limitation of the odel.Overall, the model demonstrates sufficient capability for industrial inspection across most defect types and can meet standard defect detection requirements. However, to achieve balanced and robust performance across all categories—particularly for underperforming classes like Crazing—future work should focus on strategies such as increasing sample diversity through data augmentation and refining model architecture or loss weighting to enhance detection accuracy.

In the field of steel strip surface defect detection, the limited computational resources of terminal inspection devices make it difficult to deploy deep learning–based defect detection algorithms[4]. To address this challenge, this paper proposes a lightweight steel strip surface defect detection algorithm, YOLO11. Evaluated on the NEU-DET steel strip surface defect dataset, the model achieves an mAP of 90%, delivering high detection accuracy while requiring fewer computational resources, thus demonstrating strong competitiveness in this domain. Research findings indicate

that among all defect types, crack defects on the steel strip surface have the greatest negative impact on overall detection accuracy. This is primarily due to the limited number of training samples for this class and the extreme difficulty in distinguishing such cracks from the background, which significantly increases the likelihood of both false positives and missed detections[5]. The next step is to address the challenges in detecting these types of defects and further improve the accuracy of the steel strip surface defect detection model.

List of literature:

1. LIUY,XIAOH,XUJ,etal.Arailsurfacedefectdetection methodbasedonpyramid feature and light weight convo lutional neural network[J]. IEEETransactionson Instrum entation and Measurement ,2022,71:1-10.

2. SINGHSA,DESAIKA.Automated surface defect detection framework using machine vision and convolutional neural networks[J]. Journal of Intelligent Manufacturing,2023,34 (4):1995-2011.

3. YANGAM, JIANGTY,HANY,etal.Researchonappli cation of on-linemelting in-situvisual inspectionof iron or epowder based on FasterR-CNN[J].Alex and ria Engine ering Journal, 2022, 61(11):8963-8971.

4. CAI JF,BAI J J, ZHANGX, et al.Researchon surface defect recognition of metal sheet based on improved Mask R-CNN[J]. Journal of Chongqing Institute of Science and Technology (Natural Science Edition),2023,25(2):110-116.

5. ZHOUS,ZENGY,LIS, et al.Surface defect detection of rolled steel based on light weight model[J].Applied Sciences, 2022,12(17):8905.