Sensors and Materials, Vol. 37, No. 12 (2025) 5701–5716
MYU Tokyo

S & M 4273

Detection and Classification of Automobile Wheel Hub Surface Defects Using You Only Look Once Version 8

Junyan Zhang,[1,2*] Youjie Li,[1,2] Ning Cao,[1,2] and Ruolin Ma[3]

[1]Jiangsu Union Technical Institute Suzhou Industrial Park Branch, Suzhou 215123, China
[2]Suzhou Industrial Park Industrial Technology School, Suzhou 215123, China
[3]Suzhou Early Childhood Education College, Suzhou 215123, China

(Received November 18, 2025; accepted December 15, 2025)

Keywords: hub inspection, surface defect classification, YOLOv8, CA, automatic sorting

To address the challenges of high misdetection rates, false positives, low classification efficiency, and limited adaptability to automated production lines for automobile wheel hubs, we developed a defect detection and classification algorithm based on the You Only Look Once Version 8 (YOLOv8) algorithm. The developed algorithm was deployed in a desktop-grade industrial collaborative robot, the MG400 robot (Shenzhen Dobot Corp Ltd., China), which contains a combination of 2D and 3D cameras with a calibrated ring light source to ensure a defect-to-background contrast ratio greater than 30:1. We collected images from a production line and constructed a dataset comprising microcracks, scratches, dents, protrusions, stains, and deformations. Using contrast enhancement, rotation, and flipping, the original 958 images were augmented to 1870. To improve sensor-driven defect detection, we modified the YOLOv8 algorithm by integrating the coordinate attention mechanism into the backbone module to enhance the spatial localization of defects, particularly small microcracks under low-contrast conditions. Additionally, a windowed self-attention structure from the Swin Transformer was embedded in the neck module to extract multiscale features and improve their integration. Focal loss was used in the classification loss function to mitigate model bias arising from class imbalance in the dataset. The improved YOLOv8 model showed an average detection accuracy of 98.33%, a recall of 98.89%, and a mean average precision at the intersection over union threshold of 0.5–0.95 of 99.1% on the test dataset, demonstrating significant improvements over the original YOLOv8 algorithm. The processing time per image was 45 ms, satisfying the production line requirements. In the production line verification for 1000 hubs, the classification accuracy rate of the robot with the improved algorithm reached 98.7%, with a missed inspection rate of 1.3%, and the average classification time was 0.92 seconds per product. The robot system's performance was stable despite environmental fluctuations, significantly improving the sensitivity for microcrack detection compared with traditional manual inspection methods.

---

[*]Corresponding author: e-mail: zhangjy@sipits.cn
https://doi.org/10.18494/SAM6053

ISSN 0914-4935
https://myukk.org/

1. Introduction

Algorithms used for defect detection in various industries, such as support vector machines, K-nearest neghbors, and random forests, amplify the loss proportion of difficult images, such as those including microcracks, to improve detection sensitivity. Surface microcracks, if subjected to prolonged stress, propagate, leading to the vehicle's structural failure. Scratches and dents compromise the hub's corrosion resistance, thereby reducing its service life.

Conventional inspection and classification of surface defects of wheel hubs rely on manual methods that present limitations. First, manual inspection is prone to inspectors' fatigue and subjective decisions, resulting in a low detection rate. Manual inspection is often ineffective for detecting microcracks smaller than 5 mm, which account for around 15% of total observed defects, with a false detection rate of up to 20%, a limitation consistent with studies on human visual inspection reliability.(1) Second, the inspection process is inefficient. Inspecting a single wheel hub requires approximately 10 to 15 s, which is inappropriate for AI-driven automotive manufacturing. Third, the inspection cost is high. Training inspectors is time-consuming and labor-intensive, with workforce deficiency further exacerbating the related issue. Moreover, manual inspection necessitates protective equipment and quality documentation, which requires cost and time.

The advancement of machine learning and deep learning technologies contributes to the development of defect inspection methods of wheel hubs, using object detection algorithms. You Only Look Once (YOLO) algorithms, known for their end-to-end real-time detection capabilities, have also been widely adopted in defect inspection owing to their ability to localize and classify images. YOLOv8, developed by Ultralytics, UK, introduces the C2f module to replace YOLOv5's C3 module, to optimize gradient flow by employing the path aggregation feature pyramid network (PA-FPN) to enhance multiscale feature fusion to achieve a balance between detection accuracy and speed. Its performance has been demonstrated in the defect detection of small objects. Such advancements lead to the development of various hub defect detection methods. Liu *et al.* applied YOLOv5 to hub defect detection, achieving a mean average precision (mAP) of 89.3%, although the model exhibited limited performance in detecting microcracks.(2) Lu utilized the dual-stage (DS) cascade-region-based convolutional neural network (DS-CascadeR-CNN) model with a spatial attention mechanism to classify defects, attaining an average detection accuracy of 95.49%. The model was complex, which hindered its applicability.(3) Yang *et al.* employed binocular stereo vision and the semi-global block matching algorithm for wheel hub classification with a recognition accuracy of 98.5%.(4) Such previous methods do not meet the requirements of high accuracy, real-time performance, and accurate wheel hub defect detection in automotive manufacturing.(5)

Automated, AI-driven manufacturing necessitates advanced image processing algorithms and sensor technology to collect data. The limitations of manual inspection, particularly its low detection rate for microcracks and the inefficiency of the process, necessitate a high-fidelity, repeatable data acquisition method. In the inspection process, accurate 3D images are essential to capture the geometric complexity of wheel hub defects, including depth and precise spatial information.

To develop an advanced wheel hub detection method, we adopted an enhanced YOLOv8 algorithm and incorporated a coordinate attention (CA) mechanism and a Swin Transformer (SWT) module for feature extraction. We also employed focal loss (FL) to address class imbalance in datasets. For the image process, 3D images taken by a desktop-grade industrial collaborative robot, the MG400 robot (Shenzhen Dobot Corp Ltd., China), were used. The MG400 robot is equipped with a high-precision encoder with ±0.05 mm repeatability and real-time force/position sensing. The robot's integrated controller and the vibration suppression algorithm ensure the stability and accuracy in positioning the 3D imager precisely and consistently. Utilizing the robot, we improved the YOLOv8 deep learning architecture to develop a sensor-to-algorithm system that reliably acquires, processes, and details defect data for sophisticated, sensor-driven quality control solutions in automotive manufacturing. Through such an end-to-end control of defect detection, classification, and sorting, the developed system overcomes the limitations of traditional methods.

2. Defect Detection and Classification

2.1 System architecture

The detection and classification process of automobile wheel hub defects is structured into image acquisition, algorithmic processing, and defect classification (Fig. 1).

The image acquisition unit integrates 2D and 3D cameras with an adjustable ring light source, mounted above the conveyor belt. Such a configuration enables the simultaneous capturing of 2D texture images to identify planar defects, including scratches and stains, and 3D depth data to detect volumetric anomalies, including depressions and protrusions.(6) By employing the 3D imaging technique, the limitations of the traditional 2D systems are overcome. The ring light source, with adjustable intensity, is used to minimize reflective interference and

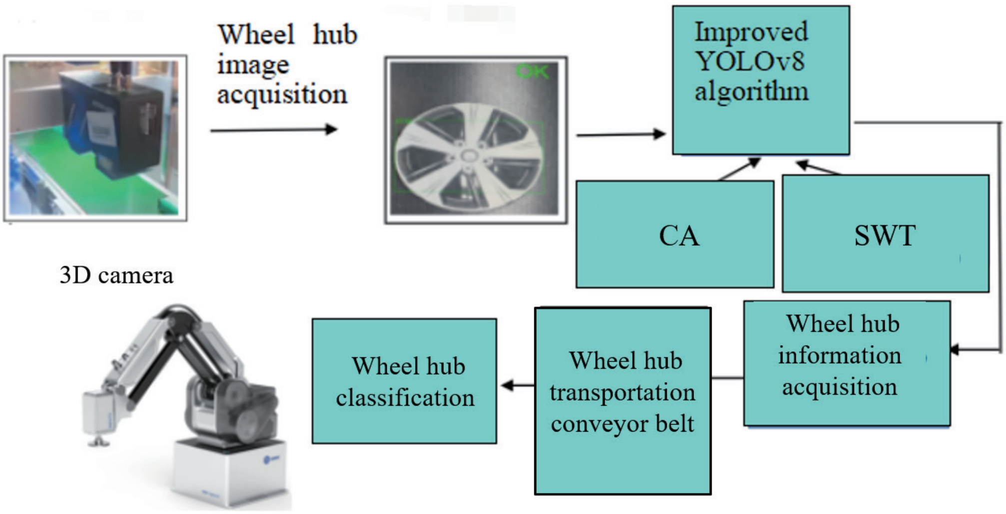

Fig. 1. (Color online) Architecture of whell hub defect detection and classification system.

ensure a contrast ratio between defect regions and background higher than 30:1, thereby enhancing defect visibility.

The algorithm processing unit features an industrial computer equipped with an Intel Core i7-12700K CPU and an NVIDIA RTX 3090 graphics processing unit (GPU).(7) It executes the enhanced YOLOv8 algorithm for defect image preprocessing, detection, and classification. Image preprocessing involves denoising and enhancing the captured images, while in defect detection, the location and nature of defects are identified. The final product's quality is determined in accordance with the classification result of defects.

The mechanical execution unit is an industrial robot with a repeatable positioning accuracy of ±0.02 mm. It communicates with the algorithm processing unit via the Ethernet for control automation technology (EtherCAT) protocol. On the basis of the received classification results, the robot actuates a flexible rubber gripper to prevent damage to the wheel hub and sorts the wheel hubs into seven bins designated for qualified products and the six distinctive defects.

Auxiliary components include a conveyor belt operating at a speed of 0.5 m/s, synchronized with the camera's image acquisition frequency. The seven designated bins are equipped with photoelectric sensors for material level monitoring and displays for the visualization of classified hubs. This architecture ensures stable and efficient system operation. The classification process is presented in Fig. 2. The following workflow is characterized by continuous acquisition, real-time processing, and precise classification.

- Hub conveyance: Wheel hubs are transported to the inspection zone via a conveyor belt. A photoelectric sensor detects their arrival and triggers an acquisition signal to the 2D and 3D cameras.
- Image acquisition: Upon signal reception, the cameras capture combined 2D and 3D imagery within 0.1 seconds and transmit the images to the computer.
- Algorithmic processing: The computer performs image preprocessing (denoising and enhancement), operates the YOLOv8 model for defect detection (completed within ≤50 ms), and outputs classification results either as qualified (no defects) or defective (in six defect types).
- Classification: The computer sends classification instructions to the robot through EtherCAT. The robot positions its grippers, classifies hubs in 0.8 s, and places the classified hub into the corresponding bins.(8)
- Status feedback: Once the bin's photoelectric sensor confirms hub placement, it sends a completion signal to the computer, initiating the next inspection process.

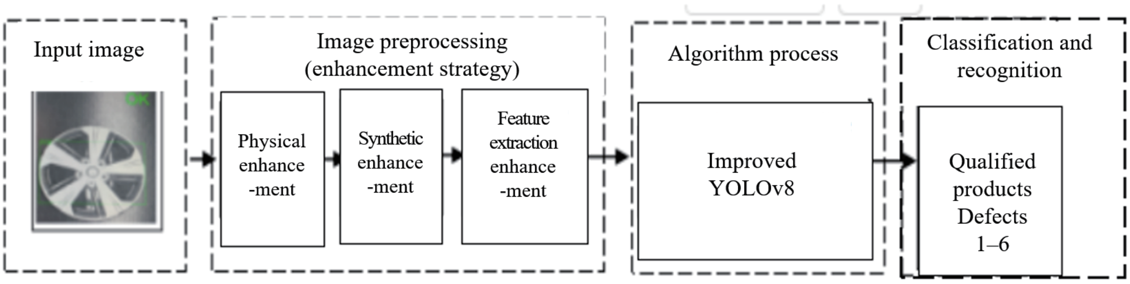

Fig. 2. (Color online) Workflow of defect classification.

2.2 Dataset collection and augmentation

The original dataset was collected from the production line of automobile wheel hub manufacturing in an anonymous company for three hub models: 18-, 19-, and 20-inch wheels.(9) 958 images containing defects were obtained, and the defects were categorized into six types (Table 1).

In the dataset, 16.9 (162) and 17.2% (165) of the images presented hubs with microcracks and stains, respectively, while there were less than 200 images with other defects. Therefore, physical and synthetic augmentation was conducted to expand the dataset to include 1870 images, ensuring a minimum of 120 samples for each defect category. The augmentation techniques used in this study included the following.

- Rotation: Random rotations between −15° and 15°, in 5° increments, were conducted to mitigate the impact of hub positional deviations.
- Horizontal flipping: With a 50% probability, hubs were simulated for inverted placements.
- Contrast adjustment: Random variation of contrast by ±20% was applied to reflect different lighting conditions.
- Gaussian denoising: Gaussian noise with a standard deviation (SD) between 0.01 and 0.03 was added to improve model robustness against image interference.
- Synthetic augmentation: Using image fusion, small regions of defects were cropped and randomly embedded into defect-free hub images, generating 120 synthetic images to bolster the representation of small defects and reduce model overfitting.

A random stratified sampling method was used to split the augmented dataset into training, validation, and test datasets in a 7:2:1 ratio to ensure balanced defect distribution between categories, minimizing the bias in model training (Table 2).

Defect regions were annotated using the LabelImg tool in the pattern analysis, statistical modeling, and computational learning visual object classes challenge (PASCAL VOC) format. Annotation guidelines specified that the smallest bounding rectangle enclosing the defect area be used as the bounding box. Images were labeled using the defect type_wheel hub model (e.g.,

Table 1
Types of wheel hub defects categorized in this study.

Defect type	Definition	Number of hubs
Microcrack	Surface crack with length ≤5 mm and width ≤0.2 mm	162
Scratch	Linear scratches with length >5 mm and depth ≤0.1 mm	185
Dent	Surface denting with depth >0.1 mm	153
Protrusion	Surface protrusion with height >0.1 mm	148
Stain	Oil stains and dust residue with an area >1 cm	165
Deformation	Slight deformation of the hub edge or spokes	145

Table 2
Dataset composition.

Sub-dataset	Total image number	Defect type	Range of number of defects
Training	1309	6	85–120
Validation	374	6	25–35
Test	187	6	12–18

microcrack_18 inches).(10) To ensure annotation precision, the intersection over union (IoU) between the bounding box and the defect area was required to be ≥0.85. Two senior quality inspection engineers conducted cross-verification of annotations, relabeling any samples with an IoU of less than 0.85 for a 100% labeling pass rate. To enhance algorithmic detection accuracy, annotated images underwent the following preprocessing.

- Gaussian filtering was applied with a 3 × 3 convolution kernel and an SD of 0.8 to suppress random noise while preserving defect features.
- Histogram equalization was conducted to expand the grayscale range from [50, 200] to [0, 255], enhancing contrast between defect regions and background.
- Image resizing was conducted to uniformly scale images to 640 × 640 pixels to meet the input requirement of the improved YOLOv8 algorithm, preventing inconsistencies in training that are caused by different image dimensions.
- Pixel normalization was conducted by dividing the pixel of an image by 255, changing the range to [0, 1] for fast model convergence.

3. YOLOv8 Algorithm

3.1 YOLOv8 architecture and enhancement

We employed a three-stage object detection framework in the YOLOv8 algorithm: backbone, neck, and head modules. Each module functions in the following distinct operations.(11)

- Backbone module (feature extraction): In the backbone, the cross-stage partial fusion (C2f) module, composed of two convolutional layers, one bottleneck layer, and residual connections, was adopted, replacing the cross-stage partial bottleneck with the three convolutions (C3) module of YOLOv5. This replacement reduced the number of parameters by 20% and improved gradient flow efficiency. The spatial pyramid pooling fast (SPPF) module was retained, aggregating feature maps using four pooling kernels of varying sizes (1 × 1, 5 × 5, 9 × 9, and 13 × 13) to enhance multiscale target adaptability. The output consisted of feature maps in three scales: prediction layer 3 (P3) (80 × 80), P4 (40 × 40), and P5 (20 × 20).
- Neck module (feature enhancement): The neck module adopts the PA-FPN structure to foster cross-scale feature interaction. It enriches low-resolution feature maps in P4 and P5 with high-resolution details from P3 through a combination of up-sampling (P5→P4→P3), down-sampling (P3→P4→P5), and convolutional fusion. This process yields fused feature maps (P3, P4, and P5) with enhanced semantic and spatial information for the detection head.
- Head module (detection output): The head module comprises classification and regression branches in its decoupled design. The classification branch predicts class probabilities in two convolutional layers, whereas the regression branch outputs bounding box coordinates in the layers. An anchor-free approach was adopted in the module, utilizing a Task-Aligned Assigner for sample matching to eliminate the need for predefined anchor boxes and simplify training. Without using predefined anchor boxes, object locations and sizes are predicted from feature maps, simplifying the detection process and improving flexibility. The loss

function was used by integrating varifocal loss (VFL) for classification and a combination of distribution focal loss (DFL) and complete IoU (CIOU) loss for regression, improving defect localization and classification accuracy.

3.2 Improved wheel hub defect detection

To detect small defects and enhance attention to defect regions, and minimize model bias due to class imbalance, the following methods were applied in the algorithm.

- CA mechanism: The C2f module augments the defect image by the CA mechanism to improve spatial attention on defect areas. The CA mechanism decomposes channel attention into one-dimensional encodings along the width and height axes.(12)
- Global feature aggregation: Feature maps [height (H) × width (W) × channel (C)] undergo average pooling in width and height to produce vectors of size $1 \times W \times C$ and $H \times 1 \times C$.
- Feature encoding: These vectors are concatenated into a $(H + W) \times 1 \times C$ tensor, compressed to C/r\$ (where r = 4 as the reduction ratio) through a 1 × 1 convolution, followed by batch normalization and rectified linear unit activation.
- Attention weight generation: Two parallel 1 × 1 convolution layers restore the channel count to C, generating directional attention weights.
- Weight application: The attention weights generated for the width and height dimensions are each expanded to match the full size of the feature map ($H \times W \times C$). The attention weights for the width and height directions are expanded to match the full spatial and channel dimensions of the feature map. These expanded weights are then combined (typically multiplied together) and applied to the original feature map via element-wise multiplication. This operation modulates the original features, enhancing those that correspond to defect regions and suppressing irrelevant background information. This step allows the algorithm to focus more precisely on defect areas such as microcracks or scratches, improving detection accuracy without significantly increasing computational cost. This attention mechanism enhances defect localization while introducing only a 0.5% increase in computational complexity, maintaining real-time performance.
- SWT integration: The C2f module in the neck module is replaced with an SWT module to enhance multiscale feature fusion.
- Windowed multihead self-attention (W-MSA): Feature maps are divided into 7 × 7 windows, and self-attention is computed locally, reducing complexity from $O(H^2 \times W^2 \times C)$ to $O((H / W_{win})^2 \times (W^4_{win} \times C))$. Here, O describes how the computation scales with input size.
- Shifted window (SW)-MSA: Adjacent layers use offset window origins [e.g., (0,0) and (3,3)] to enable cross-window interaction and mitigate boundary fragmentation.(13)
- Hierarchical output: Through four downsampling stages, the SWT module produces multiscale feature maps aligned with P3, P4, and P5, enhancing detection of defects across scales.
- FL for classification: To counteract the bias toward background samples caused by data imbalance, the original VFL is replaced with FL. Focal loss is used to address class imbalance by focusing training on well-classified samples and hard, misclassified samples. It modifies

the standard cross-entropy loss by introducing a modulating factor and a class weight. The improved YOLOv8 structure (Fig. 3) incorporates the CA-enhanced C2f module (C2F-CA) in the backbone module, the SWT module in the neck module, and FL in the head module. All other components remain consistent with the original architecture. The FL is calculated using

$$FL(pt) = -\alpha_t(1 - pt)^{\gamma} \log(pt). \tag{1}$$

Here, $(1 - pt)^{\gamma}$ represents the modulating factor that reduces the loss contribution from well-classified examples, pt is the predicted probability for the true class label, α_t is the class weight to balance the impact of each class (e.g., 0.75 for defect class, 0.25 for background), and γ is the focusing parameter (typically set to 2).

If the model is confident and correct, pt is close to 1. When pt is close to 0, the difficulty of the correct sample, α_t is set to 0.75 for defective samples (minority class) and 0.25 for background samples (majority class) to balance the category distribution. γ represents the difficulty weight, set to 2. The algorithm amplifies the loss proportion of difficult images, such as those including microcracks, to improve detection sensitivity.

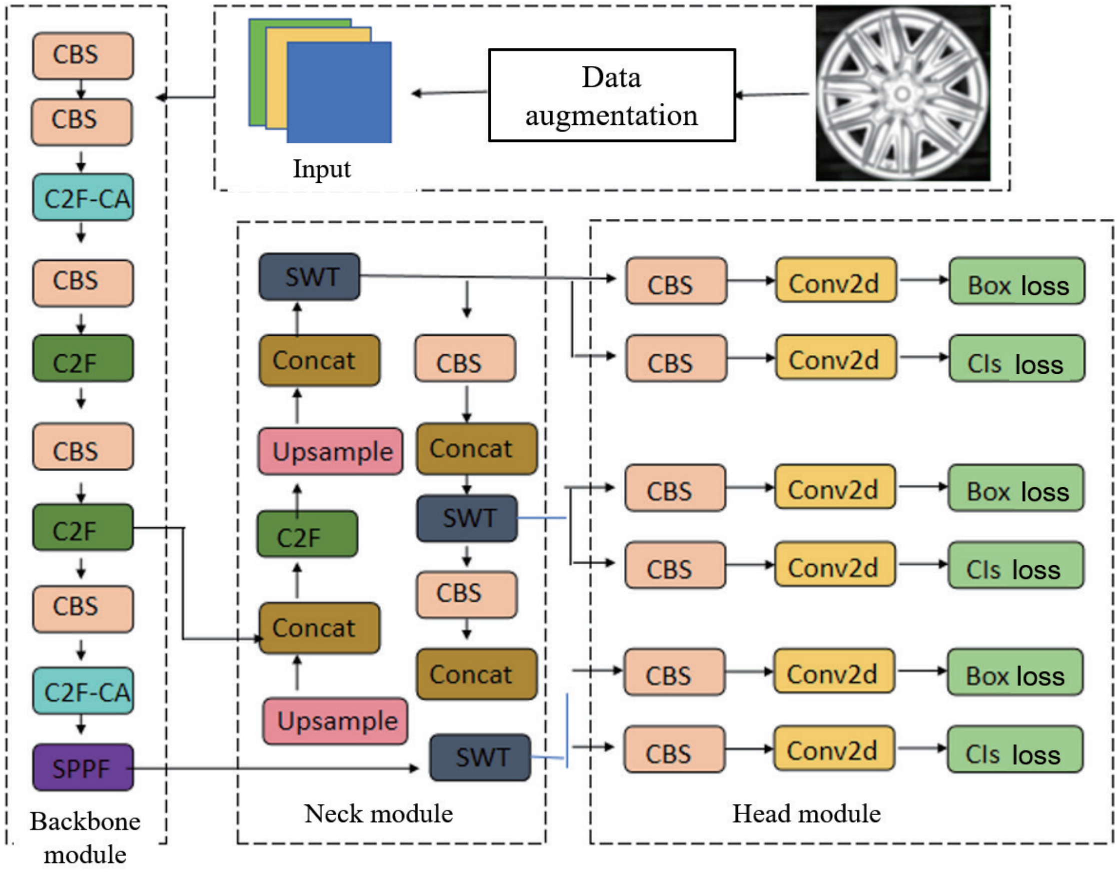

Fig. 3. (Color online) Architecture of improved YOLOv8 algorithm in this study. (CBS: convolution + BatchNorm + Sigmoid linear unit, Concat: concatenation, Upsample: upsampling, Conv2d: 2D convolution, Box loss: Bounding box loss, Cls loss: classification loss.)

4. Algorithm Training and Performance Evaluation

4.1 Hardware and software configuration

The YOLOv8 algorithm was trained and evaluated using a computer featuring an Intel Core i7-12700K CPU, an NVIDIA RTX 3090 GPU, and 32 gigabytes (GBs) of Double Data Rate Fourth-Generation random access memory (DDR4 RAM) for learning tasks. The algorithm was run on the Windows 10 operating system, employing Python 3.8 as the programming language. The deep learning framework was PyTorch 1.13.1, accelerated by Compute Unified Device Architecture 11.7. Supplementary libraries included OpenCV 4.6.0 for image processing operations and TensorBoard for comprehensive visualization and monitoring of the training process.

For stable convergence and optimal performance, training parameters were carefully selected using controlled variables. The batch size was 6 to balance the training speed and the memory constraints of the 24 GB GPU. The training began with an initial learning rate of 0.001.(14) A cosine annealing strategy was implemented to gradually reduce the learning rate to a minimum of 0.00001, which is essential for preventing the algorithm from overfitting to the training data. The algorithm was trained for 100 epochs. The first 20 epochs served as a warm-up phase, during which the learning rate was linearly increased from 0.0001 to 0.001, followed by 80 epochs for formal training. The AdamW optimizer was employed with a weight decay of 0.0005 and a momentum of 0.9. Further regularization was applied using Dropout with a probability of 0.1, combined with L2 weight decay, to actively mitigate overfitting.

The algorithm was trained using iterative methods for efficient and accurate optimization. Images and their annotations were loaded in batches using a DataLoader configured with multi-threading (utilizing num_workers = 8) and augmented by random flipping and rotation.(15) The preprocessed images were fed into the improved YOLOv8 algorithm to conduct forward propagation. The total loss was calculated as a weighted sum with a 1:1 ratio of the individual loss components. The FL function computed the classification loss to effectively address class imbalance. The regression loss for bounding box accuracy was a combination of DFL and CIOU. Gradients were automatically computed by PyTorch's Autograd mechanism, and the algorithm's parameters were updated using the AdamW optimizer.

Model performance was evaluated every five epochs using mAP and accuracy. To prevent unnecessary training, an early stopping approach was implemented. If no performance improvement was observed over ten consecutive epochs, training was terminated, and the weights corresponding to the optimal performance were stored.

4.2 Performance evaluation

To evaluate the performance of the YOLOv8 model, three comparative experiments were conducted.

In Experiment 1, the detection performance of the improved YOLOv8 model was compared with those of the original YOLOv8, YOLOv5, and DS-CascadeR-CNN using the same dataset.

For the comparison of performance, precision, recall, mAP for the IoU threshold of 0.5 (mAP@0.5), mAP for the IoU threshold of 0.5–0.95 (mAP@0.5–0.95), and average processing time per image were used. In Experiment 2, the contributions of the CA mechanism, the SWT module, and FL to overall algorithm performance were compared in an ablation study. We experimented with the original YOLOv8, YOLOv8 + CA, YOLOv8 + CA + SWT, and the improved YOLOv8 (CA + SWT+ FL). In Experiment 3, the algorithm's applicability was assessed. The algorithm was deployed in a real production line to perform continuous hub defect detection. Overall classification accuracy, missed detection rate, false detection rate, and average sorting time were calculated to assess the algorithm's applicability in industrial applications.

The evaluation metrics were calculated as follows.

1. Precision: Precision is the proportion of correctly identified defects among all defects and indicates the false positive rate.(16)

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

Here, TP represents true positives (correctly detected defects) and FP represents false positives (normal areas mistakenly detected as defects).

2. Recall: Recall is the proportion of correctly detected defects to the actual number of defects, reflecting the algorithm's missed detection rate.

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

Here, FN represents false negatives (actual defects that are not detected).

3. mAP: The area under the precision–recall curve (AP) is calculated for each type of defect, and then the average of AP for all types is taken as mAP. mAP@50 and mAP@50−95 reflect the detection accuracy of the algorithm.
4. Classification Accuracy: The proportion of correctly identified wheel hub defects by the algorithm relative to the total number of actual defects, reflecting its overall classification capability.
5. Processing/classification time: The time taken from the input of a single image to the output of detection results (processing time) and the time taken from the entry of a single wheel hub into the detection area to the completion of sorting (sorting time) are calculated to evaluate the real-time performance of the system.

5. Results and Discussion

5.1 Comparison of performance

The performances of different defect detection algorithms on the test dataset, as shown in Table 3, were compared. The superior performance of the improved YOLOv8 model across multiple metrics was observed, validating its performance enhancement.

Table 3
Performances of different algorithms on the test dataset.

Algorithm	Accuracy (%)	Recall (%)	mAP@50 (%)	mAP@50−95 (%)	Processing time per image (ms)
YOLOv5	92.15	90.38	93.2	91.5	42
Original YOLOv8	95.27	94.62	96.8	95.3	38
DS-CascadeRCNN	96.13	95.49	97.5	96.1	125
Improved YOLOv8	98.33	98.89	99.1	99.1	45

The improved YOLOv8 algorithm showed superior detection capability, yielding an accuracy of 98.33%, a recall of 98.89%, an mAP@50 of 99.1%, and an mAP@50−95 of 99.1%. The improved YOLOv8 algorithm presented substantial gains across all metrics, compared with the original YOLOv8 algorithm. Accuracy increased by 3.06% (98.33 and 95.27%), recall increased by 4.27% (98.89 and 94.62%), mAP@50 improved by 2.3 % (99.1 and 96.8%), and mAP@50−95 increased by 3.8% (99.1 and 95.3%). This substantial improvement in recall highlights the algorithm's enhanced detection ability of small targets, such as microcracks, addressing the limitation stated by Liu *et al.*(2) The performance difference from the YOLOv5 algorithm was more evident with a higher accuracy, recall, mAP@50, and mAP@50−95 by 6.18, 8.51, 5.9, and 7.6%. While YOLOv5 is known for its speed, its lower accuracy (92.15%) and mAP were insufficient to meet the quality control standards of automotive manufacturing. Although the DS-CascadeRCNN algorithm was complex in the dual-stage process, it achieved a high accuracy of 96.13% and a recall of 95.49%.(3) The single-stage improved YOLOv8 algorithm showed higher accuracy, recall, mAP@50, and mAP@50−95 by 2.2, 3.4, 1.6, and 3.0% than the DS-CascadeRCNN algorithm.

The superior performance of the improved YOLOv8, especially in mAP@50−95, addresses the issue related to the lower accuracy of previous methods.(4) The performance gains are attributed to the integrated CA mechanism, which pinpoints defect features, and the SWT module, which captures long-range dependences, whereby the feature representation for small and complex defects is improved. The use of FL minimizes the class imbalance inherent in the dataset, ensuring consistent learning across all defect types.

Considering that the requirement for the industrial deployment of the defect detection algorithm is a balance between accuracy and speed,(3,5) the advantage of the improved YOLOv8 algorithm is significant. The algorithm needed a processing time of 45 ms per image. This is a marginal increase compared with the original YOLOv8 algorithm's 38 ms. The extended processing time is attributed to the computational complexity introduced by the SWT module and the CA mechanism for deep feature extraction. However, the enhanced accuracy and mAP@50−95 can supplement the increased processing time. Its processing time (45 ms) is considerably lower than that of DS-CascadeRCNN (125 ms), proving the superiority of a single-stage detector in applications.(2) Within the 45 ms processing time, 22 images can be processed in one second, which is faster than the one image per second, which is the industrial requirement for inspecting a single wheel hub. This high-speed performance, combined with the 99.1% mAP@50−95, signifies that the improved YOLOv8 algorithm can be used for precise wheel hub defect detection in smart automotive manufacturing.(5) The improved algorithm overcomes the limitations of the conventional manual inspection process speed of 10 to 15 s, aligning with the demands of AI-driven production lines.

5.2 Ablation experiment results

Table 4 presents the ablation experiment results on the evaluation dataset, showing the contributions of CA, SWT, and FL to the overall performance of the improved YOLOv8 algorithm.(17)

By integrating the CA mechanism into the YOLOv8 model, the performance was significantly enhanced. The accuracy increased from 95.27 to 96.85%, and the mAP@50–95 improved from 95.3 to 97.2%. These results indicate that CA is the influential component for improving localization and precision. By enabling the algorithm to focus on key features and their spatial coordinates, CA enables the algorithm to effectively distinguish small defects such as microcracks. This enhancement incurred only a minimal increase of 2 ms in processing time.

The integration of the SWT module led to further performance enhancement. Accuracy increased to 97.92%, while mAP@50–95 reached 98.5%, improving by 1.3%. Recall increased from 96.15 to 97.58%. These improvements with the SWT module demonstrate that SWT strengthens feature extraction capabilities by capturing long-range dependences. This ability is crucial for detecting small, low-contrast defects that are overlooked by the convolutional backbone alone. The addition of the SWT module resulted in an increase of 3 ms in processing time.

Replacing the standard loss function with FL also contributed to performance enhancement. The improved YOLOv8 algorithm had an accuracy of 98.33%, a recall of 98.89%, and a mAP@50–95 of 99.1%, showing increases of 0.41, 1.31, and 0.6%, compared with the original YOLOv8 without using FL. The most significant enhancement was observed in recall, suggesting that FL effectively mitigates the class imbalance in the dataset. By down-weighting the loss contribution from easily classified background and large defects, FL enables the algorithm to learn hard-to-detect instances, such as rare microcracks, thereby maximizing the true positive rate.

The improved YOLOv8 algorithm, incorporating CA, SWT, and FL, achieves an mAP@50–95 of 99.1%, presenting a 3.8% improvement over the original YOLOv8 algorithm (95.3%). The ablation experiment results confirm the effectiveness of the synergistic integration of feature enhancement modules (CA and SWT) with loss function optimization (FL) on defect detection in industrial wheel hub defect inspection.(18–20)

5.3 Verification of applicability of improved YOLOv8 algorithm

Figure 4 presents defect images for six representative defect types using the improved YOLOv8 model. The images demonstrate the algorithm's ability to accurately identify and

Table 4
Ablation experiment results.

Algorithm	Accuracy (%)	Recall (%)	mAP@50–95 (%)	Processing time per image (ms)
Original YOLOv8	95.27	94.62	95.3	38
Original YOLOv8 + CA	96.85	96.15	97.2	40
Original YOLOv8 + CA + SWT	97.92	97.58	98.5	43
Improved YOLOv8 (CA + SWT + FL)	98.33	98.89	99.1	45

classify various defects, including microcracks as small as 3 mm in length and depressions as shallow as 0.1 mm in depth. These findings confirm the model's precision and robustness in detecting a wide range of surface anomalies. The robot system with the algorithm presents a lower missed detection rate, higher efficiency, and lower detection cost than traditional manual inspection methods. The algorithms for defect detection can increase the sensitivity of microcrack detection by 40% (compared with manual inspection) and classification efficiency by over 10 times, as a feasible solution for automated automobile wheel hub inspection.(19)

The improved YOLOv8 algorithm was verified in an actual production line, implemented in a desktop-grade industrial collaborative robot, the MG400 robot, running 1000 consecutive hub detection and classification tasks. The test batch of the robot system included 1000 wheel hubs, 620 qualified hubs, and 380 defective hubs. The results are summarized in Table 5, showing the algorithm's applicability to industrial applications.

The system demonstrated a classification accuracy of 98.7%, successfully classifying 987 pieces out of 1000. Crucially, the algorithm achieved a missed inspection rate of 1.3% (5 of 380),

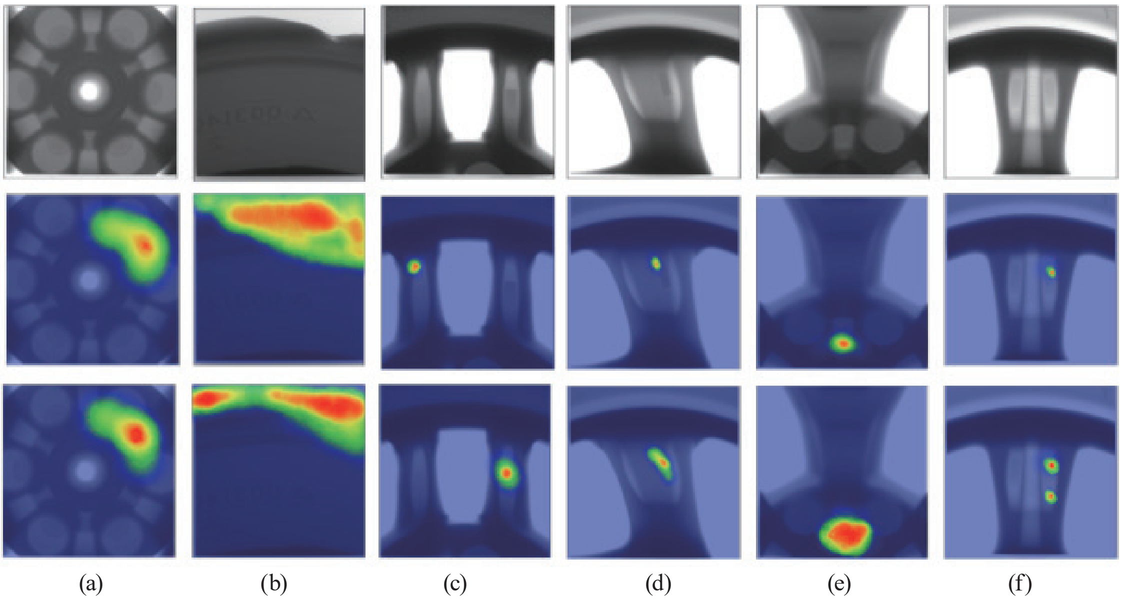

Fig. 4. (Color online) Examples of defects detected by improved YOLOv8 algorithm: (a) microcrack, (b) scratch, (c) dents, (d) protrusions, (e) stains, and (f) deformations.

Table 5
Verification results of improved YOLOv8 algorithm in MG400 robot.

Evaluation metric	Result
Total number of hubs	1000 (including 620 qualified products and 380 defective products)
Number of correctly classified defects	987 pieces
Classification accuracy	98.7%
Number of missed inspections	5 microcracks
Missed inspection rate	1.3% (5 of 380)
Number of FP	8 (stains mistakenly identified as scratches)
Detection rate	2.1% (8 of 380)
Average classification time	0.92 s per product

which is significantly better than the acceptance threshold of 2.0% for the automated systems production line used for verification in this study. All five missed inspections were microcracks with a length of less than 2 mm, representing the most challenging type of defect. Simultaneously, the system maintained a low FP rate of 2.1% (8 of 380), which is well below the industrial tolerance of "no more than 3%." The eight false positives were attributed to stains on the wheel hub surface that were mistakenly identified as scratches owing to the similarity between the stain shapes and actual scratch features, a common challenge in industrial vision systems. Overall, the algorithm meets the requirements for industrial quality inspection of missed and false detections.

The system also demonstrated excellent real-time compliance. The average classification time was 0.92 s per product. This is below the production line's time requirement of "1 second per item," confirming that the high-speed processing of the improved YOLOv8 algorithm is in the complete end-to-end robotic detection and classification cycle. This ensures the potential for continuous and stable operation without creating a bottleneck in the manufacturing process.

The system exhibited appropriate environmental adaptability and stability. During the verification test, the system maintained stable performance and high sorting accuracy despite variations in the factory, including fluctuations in the light intensity of ±20% and changes in conveyor belt speed of ±10%. This resilience confirms that the integrated system, combining the high-resolution 3D camera and calibrated light source with the robust image preprocessing and the enhanced algorithm, effectively overcomes environmental interference and can be reliably deployed in an automotive manufacturing setting.(20)

6. Conclusions

We improved a YOLO8v algorithm for the sensor-to-algorithm process and its robot system for detecting and classifying surface defects on automobile wheel hubs using 3D camera-captured images. Through experimental verification and analysis, the developed algorithm and its robot system capture high-fidelity depth and texture images utilizing a 3D camera and an adjustable ring light source, successfully resisting environmental interference. The system maintained stable performance despite fluctuations in light intensity of ±20% and conveyor belt speed of ±10% in the factory setting. The introduction of the CA mechanism in the backbone module, the SWT module in the neck module, and FL in the head module significantly enhances the accuracy of hub defect detection. Compared with the original YOLOv8, the accuracy increased by 3.06%, the recall increased by 4.27%, and the mAP@50−95 increased by 3.8%, while maintaining a processing speed of 45 ms per image, successfully balancing high accuracy and real-time performance. The end-to-end system architecture of 3D camera + improved YOLOv8 + MG400 robot was adopted to achieve the full-process automation of automatic defect detection and classification. In the production line verification of 1000 hubs, the robot system achieved a classification accuracy of 98.7%, a low missed detection rate of 1.3%, a false detection rate of 2.1%, and an average classification time of 0.92 seconds per product. This performance validates the system's robust applicability and meets the requirements of industrial production. The use of advanced sensor technology (3D camera) and the improved deep learning algorithm effectively addresses the issues of high missed detection rate, low efficiency, and high cost in traditional manual inspection. The system improves the sensitivity of microcrack detection compared with manual inspection and increases classification efficiency, providing a feasible solution for the intelligent upgrade of automobile wheel hub inspection production lines.

Although the present study results demonstrate the effectiveness of the improved YOLOv8 model, further optimization is required. At present, the algorithm is capable of detecting and classifying six types of defect. Therefore, material-specific characteristics and defect propagation patterns need to be included for categorizing microcracks into minor, moderate, and severe levels. Additionally, a deep-learning-based model needs to be developed to predict defect expansion over time, facilitating lifecycle management encompassing detection, grading, and life prediction. While the improved YOLOv8 model satisfies real-time performance requirements, it remains dependent on high-performance GPUs. To reduce hardware costs and improve deployment flexibility, model compression techniques, such as pruning and 8-bit integer quantization, need to be introduced, enabling deployment on edge computing platforms such as the NVIDIA Jetson Xavier NX. To enhance adaptability and scalability, metalearning techniques must be employed to generalize the algorithm’s ability across diverse hub types, thereby improving its versatility and expanding its applicability.

Acknowledgments

This research was supported by the Suzhou Artificial Intelligence Platform Project, specifically, the Intelligent Perception and Computing High-Quality Innovation Platform (Project No. YZCXPT2023103), and by the 2024 Planned Research Program of the China Vocational Education Association. The latter was conducted under the framework of the 'Double High Plan' and focused on exploring industry–education integration through school–enterprise cooperation and joint development (Project No. ZJS2024YB30).

References

1 J. E. See: Hum. Factors **57** (2015) 1427. https://psycnet.apa.org/doi/10.1177/0018720815602389
2 F. Liu, Z. Zhang, J. Xu, H. Zheng, and Y. Liu: Acta Meteorol. Sin. **45** (2024) 806. http://jlxb.china-csm.org:81/Jwk_jlxb/EN/Y2024/V45/I6/806
3 J. Lu: Master's Degree Thesis of Yanshan University (2022). https://doi.org/10.27440/d.cnki.gysdu.2022.001396
4 G. Yang, Y. Wang, and L. Chen: Vib. Proced. **22** (2019) 140. https://doi.org/10.21595/vp.2019.20562
5 J. Zhang, X. Sun, C. Yang, B. Wu, and P. Cao: Proc. 2025 IEEE 5th Int. Conf. Electronic Technology, Communication and Information (IEEE, 2025) 1231. https://doi.org/10.1109/ICETCI64844.2025.11084015
6 Z. Liu, H. Xu, X. Zhu, C. Li, Z. Wang, Y. Cao, and K. Dai: Comput. Eng. Sci. **45** (2024) 1444. http://joces.nudt.edu.cn/EN/abstract/abstract17966.shtml
7 X. Lyu, X. Yang, and Z. Cao: Meas. Technol. **47** (2024) 100. http://emt.cnjournals.com/emten/article/abstract/e2415797
8 Y. Gong, K. Zhang, B. Shen, H. Zhu, T. Li, N. Huang, and W. Ding: PREPRINT. https://doi.org/10.21203/rs.3.rs-7508344/v1
9 Z. Zhang, Y. Liu, and F. Liu: Acta Meteorol. Sin. **44** (2023) 1375. http://jlxb.china-csm.org:81/Jwk_jlxb/CN/abstract/abstract2546.shtml
10 H. Zhang and Y. Wu: CJSI **44** (2023) 1. https://chn.oversea.cnki.net/kcms/detail/detail.aspx?filename=YQXB202308001&dbcode=CJFQ&dbname=CJFDLAST2023&uniplatform=NZKPT

11 A. Wang, H. Chen, L. Liu, K. Chen, Z. Lin, J. Han, and G. Ding: arXiv:2405.14458v2. https://doi.org/10.48550/arXiv.2405.14458

12 I. Chen, S. Kao, H. He, W. Zhuo, S. Wen, and C.-H. Lee: Proc. 2023 IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR, 2023) 12021. https://doi.org/10.1109/CVPR52729.2023.01157

13 D. Ouyang, S. He, G. Zhang, M. Luo, H. Guo, and J. Zhan: Proc. 2023-2023 IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP., 2023) 1. https://doi.org/10.1109/ICASSP49357.2023.10096516

14 Y. Chen, C. Zhang, B. Chen, Y. Huang, Y. Sun, C. Wang, X. Fu, Y. Dai, F. Qin, Y. Peng, and Y. Gao: Comput. Biol. Med. **170** (2024) 107917. https://doi.org/10.1016/j.compbiomed.2024.107917

15 Y. Yu, M. Wang, Z. Wang, and P. Zhou: Proc. 2021 IEEE 4th Advanced Information Management, Communications, Electronic and Automation Control Conf. (IMCEC, 2021) 1386. https://doi.org/10.1109/IMCEC51613.2021.9482386

16 C. Tang, X. Feng, H. Wen, X. Zhou, Y. Shao, X. Zhou, B. Huang, and Y. Li: Appl. Sci. **11** (2021) 10508. https://doi.org/10.3390/app112210508

17 Github: https://github.com/ultralytics/ultralytics (accessed November 2025).

18 W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg: Proc. ECCV Lecture Notes in Computer Science 2016, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. (ECCV, Amsterdam, 2016) 9905. https://doi.org/10.1007/978-3-319-46448-0_2

19 S. Ren, K. He, R. B. Girshick, and J. Sun: IEEE Trans. Pattern Anal. Mach. Intell. **39** (2015) 1137. https://doi.org/10.1109/TPAMI.2016.2577031

20 Y. Song, L. Lu, G. Wang, and Y. Wang: Proc. 2022 Int. Conf. Mechanisms and Robotics (ICMAR, 2022) 123310X. https://doi.org/10.1117/12.2652519