Abnormal State Detection of Industrial Tools Based on the MGC-YOLOv8 Algorithm

Guan Yang[1], Xiang Cheng[2], Miao Wang[3]*, Ziyue Huang[4], Hao Tang[5], Yujun Chen[6]
School of Artificial Intelligence, Zhongyuan University of Technology, Zhengzhou, China[1,2,4]
School of Computer Science, Zhongyuan University of Technology, Zhengzhou, China[2,4,5]
School of Software, Henan University of Engineering, Zhengzhou, China[3]
Faculty of Engineering, Architecture and Information Technology, The University of Queensland, Brisbane, Australia[6]

***Abstract*—As intelligent manufacturing advances toward precision and automation, cutting tool condition critically impacts product quality, equipment safety, and production efficiency. Anomalies like wear, chipping, or fracture cause workpiece scrapping and machine failure, demanding efficient online monitoring. Traditional manual or image-based methods suffer from low accuracy in complex environments. Although deep learning excels in industrial defect detection, existing end-to-end detectors exhibit insufficient recall and localization precision for millimeter-scale cracks and blurred tool boundaries. To address these challenges, we propose MGC-YOLOv8, an enhanced framework built upon the YOLOv8 backbone. A Multi-Scale Edge-Dual Fusion (MSEDF) module is introduced to integrate feature maps across different scales, thereby strengthening the detection of minor defects. Furthermore, a Global-to-Local Spatial Aggregation (GLSA) module enriches feature representations by simultaneously capturing global context and local details. A Convolutional Block Attention (CBAM) module is embedded upstream of the prediction head to adaptively highlight critical features in both channel and spatial dimensions.Although the integration of MSEDF, GLSA, and CBAM introduces a marginal runtime overhead and a slight increase in parameter count, the optimized architecture preserves real-time inference speeds that fully satisfy the requirements of industrial inspection systems. Experimental results demonstrate that MGC-YOLOv8 substantially outperforms the baseline YOLOv8n, achieving 88.1% precision, 87.9% recall, 92.5% mAP@0.5 and 69.6% mAP@0.5:0.95 on our test set.**

***Keywords*—Object detection; surface defect detection; YOLOv8**

I. Introduction

During machine-tool processing, real-time monitoring of tool condition can effectively prevent workpiece scrap and machine-tool damage [1]. Maintaining good tool condition is crucial for CNC machining quality and productivity [2], making Tool Condition Monitoring (TCM) an essential part of modern manufacturing [3]. Traditional inspection—relying on skilled operators for visual checks or intermittent sampling—is inefficient and vulnerable to subjectivity and fatigue, and thus unsuitable for large-scale continuous online monitoring [4]. Early automated TCM used force and acceleration sensors (cutting force, acoustic emission, vibration) with time-frequency analysis to infer wear [5], [6], [7], [8]. However, fluctuations in machining parameters and workshop noise often undermined their robustness, causing missed detections and false alarms.

Vision-based tool anomaly detection has emerged as a significant research focus alongside rapid advances in deep learning [9]. Convolutional neural networks (CNNs) [10] automatically extract semantic features through end-to-end training, yielding substantial gains in accuracy and real-time performance. The YOLO family adopts a single-stage framework with multi-scale feature-pyramid fusion, achieving a favorable speed–accuracy trade-off for industrial defect detection [11], [12]. However, standard downsampling and fusion in these detectors can attenuate or lose millimetre- and sub-millimetre-scale defects (fine cracks, chipping, blurred edges) under complex lighting and cluttered backgrounds, degrading recall and localization precision.

To address these issues, we propose MGC-YOLOv8. We replace the original C2f with a Multi-Scale Edge-Dual Fusion (MSEDF) module to better capture edge and fine-detail cues of subtle anomalies, enhancing local feature representation. A Global-to-Local Spatial Aggregation (GLSA) module uses a dual-branch decoupled design to extract global context and local details and fuses them via dynamic gating to improve robustness in complex scenes [13]. A Convolutional Block Attention Module (CBAM) is embedded immediately before the prediction head to adaptively reweight channel and spatial responses and focus on anomaly-sensitive regions [14], [15]. Experimental results show that MGC-YOLOv8 yields substantial accuracy gains over the YOLOv8n baseline.

The main contributions of this study are as follows:

- We designed the MSEDF to replace YOLOv8's C2f, explicitly capturing multi-scale edges and fine-detail cues of micro anomalies (e.g., cracks, wear), thereby enhancing local feature representation.
- We introduce the GLSA module, a dual-branch design that separately extracts local details and global semantics, then adaptively fuses them via dynamic gating — improving the robustness and efficiency of multi-scale feature fusion.
- We embed the CBAM before the prediction head before the prediction head, where channel–spatial attention adaptively reweights features to emphasize anomaly-sensitive regions and improve detection accuracy.

*Corresponding author.

- An industrial tool dataset with diverse real surface defects was constructed to support model training and evaluation, verifying the practicality and effectiveness of the proposed method.

The remainder of this study is organized as follows: Section II reviews related work in tool-condition monitoring and vision-based defect detection. Section III details the architecture of the proposed MGC-YOLOv8 model and the design of its key modules. Section IV describes the experimental setup, dataset construction and evaluation metrics. Section V presents the experimental results and comparative analysis. Finally, Section VI concludes the study and outlines future research directions.

II. Related Work

In recent years, tool condition detection has shifted from empirical inspection and signal analysis to image-based and deep-learning approaches. Early methods using physical signals (e.g., cutting force, vibration) with wavelet or Fourier analysis were easily affected by noise, parameter fluctuations, and nonlinear factors, leading to unstable results. Machine vision enabled direct acquisition of tool-surface images for defect identification, offering high spatial resolution but limited adaptability to complex backgrounds. With the advance of deep learning, CNN-based object detection [16]—featuring automatic feature extraction and end-to-end training—has become a key means to enhance accuracy and real-time performance, addressing the limitations of traditional techniques and driving intelligent, automated tool-condition monitoring in modern manufacturing.

Tool Condition Monitoring (TCM) can be divided into indirect monitoring based on physical signals and direct monitoring based on visual information. Early studies emphasized indirect methods, using cutting-force and vibration signals with time–frequency features to classify wear states. For example, Zhu et al. [17] reviewed wavelet transform for TCM, noting its multi-resolution ability to extract non-stationary features, but also its dependence on noise-model assumptions, redundant coefficients, and empirical thresholds, which limit high-frequency fault extraction and real-time decision-making. Zhou et al. [18] proposed vibration-based monitoring via WTMM and SVM, showing sensitivity to wear-transition stages, yet suffering from noise interference and empirical parameter settings. He et al. [19] introduced temperature-signal-based prediction using SSAEs, achieving high accuracy but relying on empirical hidden-node reduction, restricting real-time adaptability. To address small-sample challenges, Zhou et al. [20] designed MEGNN to encode cutting-force signals as recursive graphs, improving classification but at the cost of high computational complexity in multi-scale convolution and graph updates, limiting real-time use. Overall, while such signal-based methods reveal intrinsic links between signals and wear, they remain vulnerable to parameter fluctuations and environmental noise. Furthermore, the inherently low signal-to-noise ratio of acoustic emissions easily masks minor wear features, often necessitating complex denoising procedures [21].

With advances in image processing, direct monitoring using machine vision has attracted increasing attention, offering high spatial resolution and intuitive defect visualization without multi-source signal preprocessing. Guo et al. [22] employed adaptive GrabCut with improved Gaussian mixture modeling and nonlinear filtering to isolate tiny edge defects. However, this approach required manual tuning, was sensitive to illumination and background variations, and demanded machining interruption. Yu et al. [23] developed a pixel-based wear quantification approach with local-variance segmentation and image enhancement, achieving accuracy but at the cost of redundant parameters, high computation, and limited capability for sub-millimeter defects in cluttered scenes. Tang et al. [24] proposed MiSLASND, a ResNet32-based deep-transfer model with class-balance regularization, which improved accuracy and efficiency but lacked defect localization and relied on costly two-stage training. Overall, image-based TCM avoids the noise susceptibility of signal-based methods but still faces challenges in robustness, real-time applicability, and fine-scale defect detection.

Indirect monitoring via physical signals can capture tool-wear trends; however, it is highly sensitive to noise and varying conditions. Conversely, while image-based monitoring offers intuitiveness, it struggles with minor-defect detection in complex backgrounds. To address these issues, researchers have explored more efficient architectures. The YOLO (You Only Look Once) framework [25], as a single-stage detector, enhances small-target recognition by optimizing Feature Pyramid Networks (FPN) [26] and Path Aggregation Networks (PAN) [27], achieving both accuracy and speed [28]. This balance of real-time performance and precision makes YOLO highly promising for industrial defect detection, particularly in tool anomaly monitoring.

To address the limitations of traditional image-based monitoring in fine-grained feature extraction, multi-scale fusion, and robustness to complex backgrounds, we propose an enhanced YOLOv8-based method for industrial tool anomaly detection. The network integrates MSEDF for edge/detail preservation, GLSA for adaptive global–local fusion, and CBAM for channel–spatial attention, thereby improving sensitivity to minute defects and overall detection accuracy in complex industrial scenarios.

III. Methodology

A. MGC-YOLOv8

Although YOLOv8 offers high efficiency in real-time detection, it struggles with industrial tool anomalies: microcracks and slight abrasion are often lost during downsampling, while uneven illumination, glare, and cluttered backgrounds further obscure fine details. Its global fusion strategy also fails to preserve local edge and texture information, leading to false positives on sub-millimetre defects. Thus, the standard YOLOv8 falls short of the precision and robustness required in industrial applications.

As shown in Fig. 1, we propose the MGC-YOLOv8 model, which incorporates targeted enhancements to the original YOLOv8 architecture to bolster both local detail representation and multi-scale feature fusion. In the Backbone and Neck stages, we replace the standard C2f modules with our MSEDF module, which significantly improves the extraction of edge features associated with subtle anomalies—such as fine surface

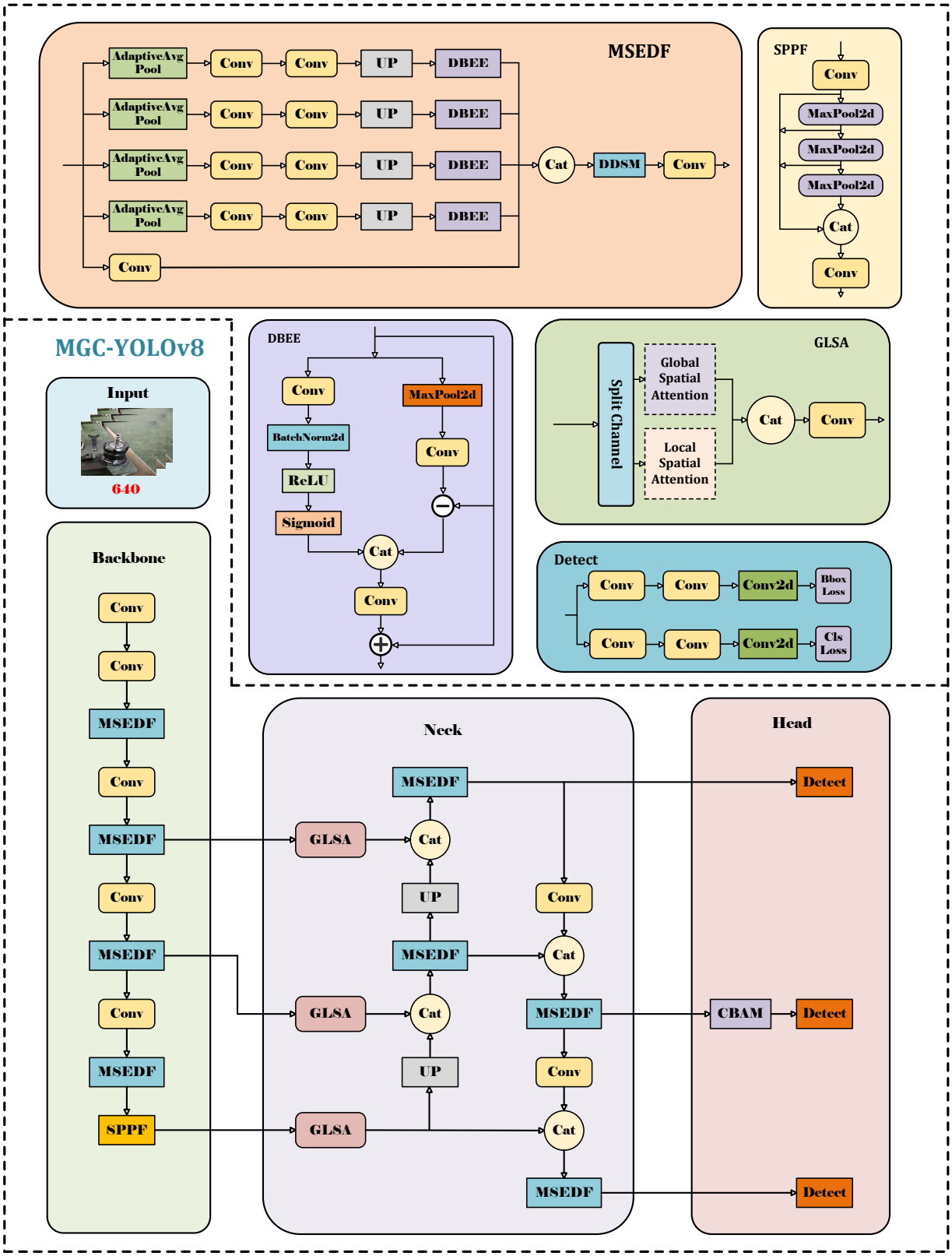


Fig. 1. MGC-YOLOv8 structure.

cracks and wear on cutting tools. Within the Neck, the GLSA module is introduced. Its dual-branch structure separately captures local detail cues and global contextual semantics. Meanwhile, a dynamic gating mechanism adaptively fuses these streams to enhance information flow across scales and improve robustness. Just before the detection head, we embed a CBAM to dynamically modulate feature responses along both channel and spatial dimensions, directing the model's focus toward critical anomalous regions and thereby further increasing detection precision. By synergistically combining MSEDF's micro-defect perception, GLSA's enriched multi-scale interactions, and CBAM's refined attention, MGC-YOLOv8 achieves a substantial improvement in accurately detecting industrial tool abnormalities.

B. Multi-Scale Edge-Dual Fusion Module

In industrial tool anomaly detection, the original C2f module in YOLOv8 exhibits clear limitations when handling local irregularities—such as cracks and surface wear—because its global feature-fusion strategy fails to capture fine edge and detail information. To overcome this, we introduce the MSEDF module, illustrated in Fig. 2. MSEDF employs adaptive average pooling to generate feature maps at multiple scales, each of which is processed by a Dual-Branch Edge Enhancer(DBEE) to strengthen edge and detail cues, thereby improving detection accuracy. Furthermore, we integrate a Dual Domain Selection Mechanism (DSM) [29], which adaptively focuses on the most task-relevant regions across these scales by selectively attending to higher-importance features. This targeted selection markedly enhances feature discriminability and boosts overall model performance.

The Dual-Branch Edge Enhancer optimizes edge features via a cooperative workflow of high-frequency residual enhancement and detail feature amplification. In the first branch, the high-frequency residual feature $\mathbf{R}$ is extracted as follows. Given an input feature map $\mathbf{X} \in \mathbb{R}^{H\times W\times c}$, a local extremum is obtained by applying MaxPool, then convolved with a 3×3 kernel $\mathbf{W}_2$. A sigmoid activation $\sigma(\cdot)$ converts the result into a spatial attention weight in $[0,1]$. Subtracting this weighted low-frequency component from the original feature yields the high-frequency residual. The calculation is shown in Eq. (1):

$$\mathbf{R} \quad = \quad \mathbf{X} - \sigma\big(\mathbf{W}_2 * \mathrm{MaxPool}(\mathbf{X})\big) \tag{1}$$

Meanwhile, the second branch extracts the fine-detail feature $\mathbf{D}$ by applying a 3×3 convolution $\mathbf{W}_3$, followed by BatchNorm and ReLU (denoted as $\varnothing(\cdot)$). The calculation is given in Eq. (2):

$$\mathbf{D} = \varnothing(\mathbf{W}_3) \tag{2}$$

The outputs of both branches are concatenated along the channel dimension and then channel-compressed via a 1×1 convolution $\mathbf{W}_1$. A residual connection adds the original feature $\mathbf{X}$ to preserve baseline information. The final enhanced edge feature $\mathbf{F}_{\mathrm{DBEE}}$ is therefore computed as shown in Eq. (3):

$$\mathbf{F}_{\mathrm{DBEE}} = \mathbf{X} + \mathbf{W}_1\left[(\mathbf{X} - \sigma(\mathbf{W}_2 * \mathrm{MaxPool}(\mathbf{X}))) \parallel \varnothing(\mathbf{W}_3)\right] \tag{3}$$

where, "$\parallel$" denotes concatenation along the channel axis.

MSEDF decomposes the input feature $\mathbf{X}$ into multi-scale space. Adaptive average pooling $\mathrm{Proj}_s(\mathbf{X}) = \mathit{AdaptiveAvgPool}_{s\times s}$ is used to generate sub-features of resolution $s \times s$ to capture the detail information of the tool surface and edge at different resolutions.

Each sub-feature is enhanced by an independent Dual-BranchEdgeEnhancer for local edge details, and then restored to the original resolution by bilinear interpolation $\mathit{Interp}_{H\times W}$ to obtain the multi-scale feature $\mathbf{F}_s^{\uparrow} \in \mathbb{R}^{C\times 1\times 1}$.

All scale features are concatenated with the local 3×3 convolution feature $\mathbf{W}_5$ into $\mathbf{F}_{\mathrm{concat}} \in \mathbb{R}^{H\times W\times 4C}$, and the dual domain selection mechanism $\mathrm{DSM}(\cdot)$ is input to dynamically balance the feature weights.

Finally, the channel is compressed through the 3×3 convolution kernel $\mathbf{W}_4$, and the optimized feature $\mathbf{Y}$ is output. The overall calculation is shown in Eq. (4):

$$\mathbf{Y} = \mathbf{W}_4 * \mathrm{DSM}\left(\left[\mathbf{W}_5 * \mathbf{X}; \bigoplus_{s\in S} \mathit{Interp}_{H\times W}\left(\mathbf{F}_{\mathrm{DBEE}}(\mathrm{Proj}_s(\mathbf{X}))\right)\right]\right) \tag{4}$$

In summary, the MSEDF module employs a multi-scale strategy to capture tool-surface details at varying resolutions and leverages its embedded DBEE to strengthen local edge and texture representations. Subsequently, an adaptive attention mechanism fuses information across these scales, thereby substantially improving the detection of minute anomalies such as micro-cracks and wear.

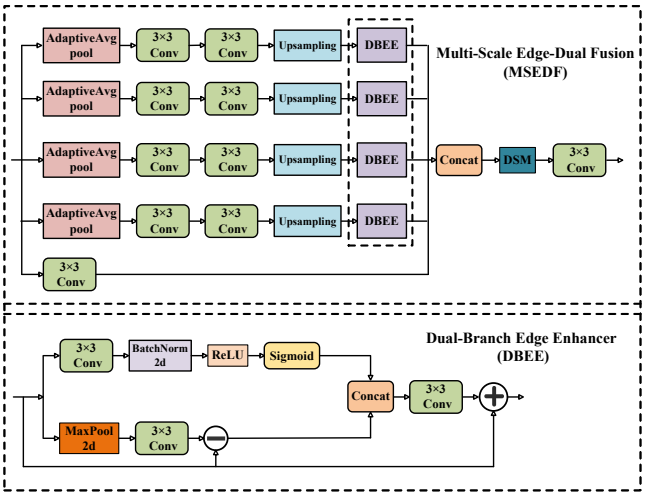


Fig. 2. Structural diagram of the MSEDF network.

C. Global-to-Local Spatial Aggregation Module

Traditional YOLOv8 suffers high false-positive rates and weakened small-object features due to repeated downsampling

and complex-background interference, hindering precise localization of chipping, micro-cracks and similar defects. To address this, a GLSA module is incorporated. GLSA adopts a dual-branch design: a local branch using depth-wise separable convolutions combined with Local Spatial Attention (LSA) to reinforce minute-defect representation, and a global branch employing Global Spatial Attention (GSA) to capture holistic scene context. A dynamic gating mechanism adaptively fuses the two streams, improving multi-scale integration and robustness to cluttered backgrounds, and thus enhancing detection accuracy for fine-grained tool defects.

As shown in Fig. 3, each multi-scale input is split along the channel dimension into two sub-features and passed to the global and local branches. In our implementation, this split ratio is fixed at 0.5 (i.e., splitting the channels equally) to maintain a balance between computational efficiency and feature representation, rather than being dynamically learned.In the global branch, channel reduction is followed by computation of a spatial attention map, which is fused with the sub-feature and processed by an MLP to capture holistic semantics. The local branch uses standard and depth-wise separable convolutions to extract fine details, produces a local spatial attention map, and fuses it with the sub-feature to strengthen local cues. Finally, both branch outputs are combined via convolution and adaptively integrated by a dynamic gating mechanism to yield the module output.

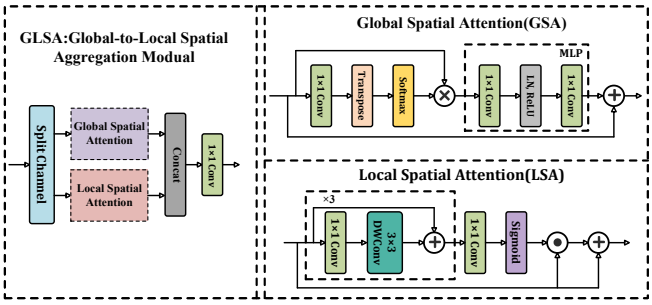


Fig. 3. GLSA module network structure diagram.

D. Convolutional Block Attention Module

Industrial tool-surface anomalies (e.g., cracks, wear) are minute, irregular, and low-contrast. Furthermore, in complex industrial scenes, uneven illumination, specular reflections, and background noise often attenuate these cues during feature extraction. To mitigate this, a CBAM is embedded immediately before the YOLOv8 detection head (see Fig. 4). By applying channel-wise and spatial attention, CBAM adaptively recalibrates feature responses to emphasize anomaly-sensitive regions and suppress irrelevant information, thereby improving detection accuracy and robustness.

The Channel Attention Module (CAM) in CBAM (see Fig. 5) aggregates channel-wise statistics using global average and max pooling, feeds both descriptors through a shared MLP, and sums them element-wise. A sigmoid activation converts the result into channel attention weights that adaptively reweight channels—amplifying informative channels for anomaly detection while suppressing irrelevant or redundant responses.

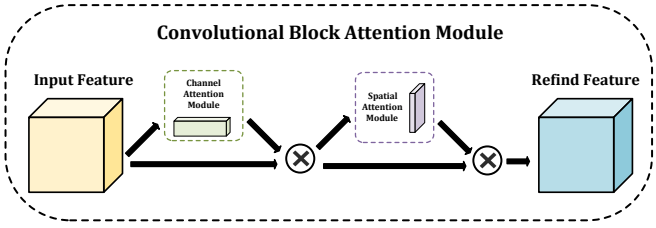


Fig. 4. Convolutional block attention module structure diagram.

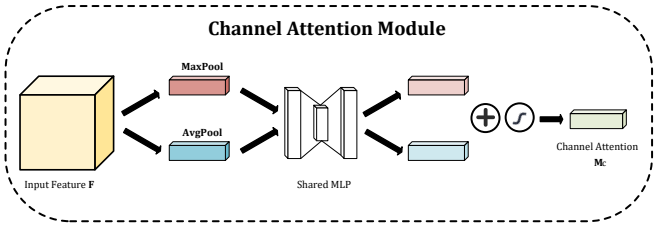


Fig. 5. Channel attention module structure diagram.

The Spatial Attention Module (SAM) processes the channel-refined feature map (see Fig. 6). It applies channel-wise average and max pooling to produce two 2D maps, concatenates them, and uses a convolution to generate a spatial attention map that highlights anomaly-prone regions and guides the network to focus on critical spatial locations.

IV. Experiment Preparation

A. Dataset Production

A private industrial dataset was collected from the production line of a factory in Yinchuan under natural operating conditions, covering different machining stages (roughing and finishing) and multiple material-cutting scenarios to ensure diversity. Cameras recorded cutting tools in four representative states—normal, chipped, broken and worn—resulting in 4,285 raw images. To guarantee data quality, a two-step screening was performed: automatic checks of resolution and file integrity, followed by manual inspection for clarity, exposure, motion blur, occlusion, and background interference. Low-quality samples were discarded, and 3,131 high-quality images were retained. All valid images were resized uniformly to

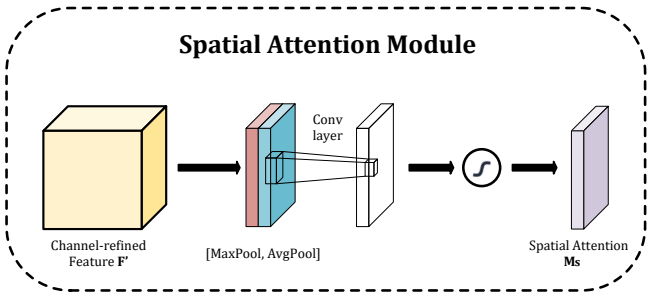


Fig. 6. Spatial attention module structure diagram.

640×640 pixels to preserve tool details while aligning with deep-learning input standards.

Annotations were produced using the LabelMe platform by technicians experienced in tool-state characteristics. Each image was annotated with bounding boxes and classified into four categories—chipped edge, broken tool, wear, and normal—as illustrated in Fig. 7. To ensure reliability, multiple rounds of review and cross-checking were conducted, with ambiguous cases resolved through discussion. Finally, the dataset was divided into training, validation, and test subsets in an 8:1:1 ratio, enabling effective parameter learning, model selection, and independent evaluation, and providing a solid basis for subsequent experiments.

Fig. 7. Sample graph of the dataset.

B. Experimental Environment

The experiments were implemented in the PyTorch framework (v2.4.1) with NVIDIA CUDA 12.2, on a hardware platform equipped with an Intel i7-10700K CPU and an NVIDIA RTX A5000 GPU. Training was performed for 300 epochs using the SGD optimizer with an initial learning rate of 0.01, a batch size of 16, and an input image resolution of 640×640. Other hyperparameters followed the official YOLOv8 settings. Since defect detection is highly sensitive to tool orientation and lighting conditions, data augmentation strategies were explicitly employed during the training phase. Specifically, we natively leveraged the robust data augmentation pipeline embedded in the YOLOv8 framework. These techniques primarily include Mosaic augmentation, random scaling, translation, and HSV color jittering. These strategies effectively expanded the diversity of the training samples, mitigating the impact of complex industrial backgrounds and varying illumination. Furthermore, to ensure a fair comparison, all competing models were evaluated under identical dataset conditions and initialized with standard pretrained weights rather than being trained from scratch.Throughout the study, a consistent software and hardware environment was maintained to ensure reproducibility.

C. Evaluation Index

To evaluate model performance in detecting tool anomalies (e.g., wear, fracture, vibration), Precision, Recall, F1-Score, and mean average precision (mAP) were used, with mAP reported as mAP@0.5 and mAP@0.5:0.95. These metrics jointly assess localization and classification accuracy, providing a quantitative basis for the reliability and applicability of tool-state detection.

Precision refers to the proportion of tool samples predicted as anomalous that are actually anomalous, reflecting the model's accuracy in identifying abnormal tool states.It is calculated as shown in Eq. (5):

$$Precision = \frac{TP}{TP + FP} \tag{5}$$

Recall measures the proportion of truly anomalous tool samples that the model successfully detects, with its calculation given in Eq. (6):

$$Recall = \frac{TP}{TP + FN} \tag{6}$$

The F1-Score, defined as the harmonic mean of precision and recall, balances the model's accuracy in predicting anomalous states with its ability to minimize missed detections. Its calculation is given in Eq. (7):

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{7}$$

Mean Average Precision (mAP) is a standard metric in object detection, defined as the mean of Average Precision (AP) values across all classes, where AP corresponds to the area under the precision–recall curve. Specifically, mAP@0.5 denotes the mean AP at an IoU threshold of 0.5, as formulated in Eq. (8):

$$mAP@0.5 = \frac{1}{C} \sum_{C=1}^{C} AP_{0.5}^{(c)} \tag{8}$$

To evaluate detection performance across different overlaps, mAP@0.5:0.95 is used, defined as the mean AP over IoU thresholds from 0.5 to 0.95, as shown in Eq. (9):

$$mAP@0.5:0.95 = \frac{1}{T} \sum_{t=1}^{T} \left(\frac{1}{C} \sum_{c=1}^{C} AP_{IoU_t}^{(c)} \right) \tag{9}$$

V. Experiment and Analysis

A. CBAM Routing Location Experiment

The Convolutional Block Attention Module (CBAM) integrates channel and spatial attention to enhance feature representations. In object detection, however, the effectiveness of such attention depends strongly on its placement within the multi-scale feature-fusion pipeline. This study therefore examines how different CBAM insertion points affect feature extraction, semantic representation, and detection performance [30].

Table I outlines the 26-layer structure of MGC-YOLOv8, where the Index column indicates each module's position, Module specifies its type, and Route denotes input-layer indices. The three detection heads take inputs from layers 18, 21, and 24. Since CBAM placement before these heads may affect performance, comparative experiments were conducted, and the results are shown in Table II.

Since the YOLOv8 detection head has only three output layers, there are seven possible CBAM insertion schemes, as detailed in Table II. The experimental results clearly show that embedding CBAM at positions (24, 25) yields the best overall performance. Accordingly, in the MGC-YOLOv8 model, CBAM is inserted at routing positions (24, 25).

TABLE I. NETWORK STRUCTURE INFORMATION OF MGC-YOLOV8

Index	Module	Route	Index	Module	Route
0	Conv		13	Upsample	12
1	Conv	0	14	Concat	13,11
2	MSEDF	1	15	MSEDF	14
3	Conv	2	16	Upsample	15
4	MSEDF	3	17	Concat	16,10
5	Conv	4	18	MSEDF	17
6	MSEDF	5	19	Conv	18
7	Conv	6	20	Concat	19,15
8	MSEDF	7	21	MSEDF	20
9	SPPF	8	22	Conv	21
10	GLSA	4	23	Concat	22,12
11	GLSA	6	24	MSEDF	23
12	GLSA	9	25	Predict	18,21,24

B. Ablation Experiments

To assess the effectiveness of our collaborative multi-module optimization, we conducted ablation studies on the YOLOv8n baseline, evaluating seven configurations: individual modules, all pairwise combinations, and the full MSEDF+GLSA+CBAM ensemble. Results are summarized in Table III.

Table IV shows that each module improves YOLOv8n performance. MSEDF alone increases precision, recall, F1-Score, and mAP metrics, while CBAM further boosts precision and F1-Score. GLSA alone provides modest gains but, combined with MSEDF or CBAM, significantly enhances performance by capturing global semantic context. Integrating all three modules achieves the best results: 88.1% precision, 87.9% recall, 88.0% F1-Score, 92.5% mAP@0.5, and 69.6% mAP@0.5:0.95, confirming their complementary roles and the effectiveness of the multi-module strategy.

C. Comparative Experiment

To validate the efficacy of the proposed MGC-YOLOv8 algorithm for industrial tool-anomaly detection, we conducted comparative experiments with several state-of-the-art object-detection frameworks, including multiple YOLO series [31], [32], [33], [34], [35], RT-DETR [36], Mamba-YOLO [37], and Hyper-YOLO [38]. To ensure statistical reliability and address potential performance variability, the evaluation metrics in Table IV are reported as the average along with the standard deviation over 3 independent runs. As shown in Table IV, MGC-YOLOv8 achieves the highest $88.1\% \pm 0.2\%$ Precision and $87.9\% \pm 0.2\%$ Recall, outperforming the next-best model, Hyper-YOLO, by about 1.5 percentage points. It also obtains the best $88.0\% \pm 0.1\%$ F1-Score and average precision, with an mAP@0.5 of $92.5\% \pm 0.1\%$ and mAP@0.5:0.95 of $69.6\% \pm 0.2\%$. The minimal variability observed demonstrates that our model's performance improvements are statistically meaningful and highly stable.

Among the baseline YOLO models, YOLOv3 showed limited accuracy in detecting fine defects. YOLOv8n improved with enhanced multi-scale fusion and bounding-box prediction, reaching $89.9\% \pm 0.3\%$ mAP@0.5, yet still lagged behind our method. Hyper-YOLO, with its advanced neck design, achieved $91.1\% \pm 0.2\%$ mAP@0.5 and $86.6\% \pm 0.3\%$ precision but was still inferior to MGC-YOLOv8 across all metrics.

Furthermore, addressing the critical concern of computational efficiency and runtime overhead in industrial inspection systems, Table IV introduces a comparative analysis of parameter counts (Params) and inference speeds (FPS). The baseline YOLOv8n achieves a high inference speed of approximately 210 FPS with 3.01M parameters. The integration of our proposed MSEDF, GLSA, and CBAM modules inevitably increases the parameter count to 3.95M, resulting in an inference speed of 165 FPS. However, this marginal runtime overhead is a highly worthwhile trade-off, as 165 FPS still far exceeds the standard real-time processing threshold required by industrial cameras. Compared to complex architectures like RT-DETR (120 FPS, 20.0M Params) and Hyper-YOLO (135 FPS, 6.5M Params), MGC-YOLOv8 offers a definitively superior speed-accuracy trade-off.

Finally, the training curves in Fig. 8 show that MGC-YOLOv8 converges more rapidly and smoothly, indicating that the integration of MSEDF, GLSA, and CBAM modules effectively improves feature extraction for subtle tool defects.

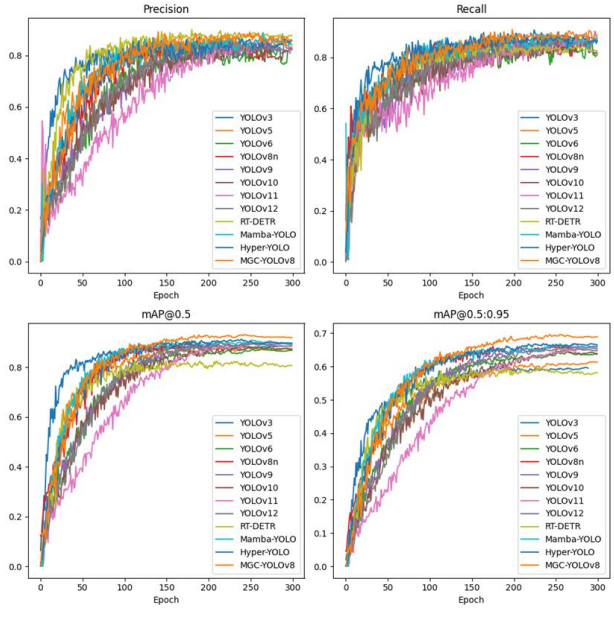


Fig. 8. Comparison of Precision, Recall, mAP@0.5 and mAP@0.5:0.95 curves trained with different algorithms.

TABLE II. CBAM ROUTING LOCATION EXPERIMENTAL RESULTS

Model	Precision	Recall	F1	mAP@0.5	mAP@0.5:0.95
CBAM (18,19)	84.50%	85.30%	84.90%	89.80%	67.60%
CBAM (21,22)	**88.10%**	**87.90%**	**88.00%**	**92.50%**	**69.60%**
CBAM (24,25)	84.20%	87.00%	85.58%	92.30%	68.10%
CBAM (18,19)(21,22)	87.70%	87.10%	87.40%	91.90%	68.70%
CBAM (18,19)(24,25)	86.70%	84.60%	85.64%	91.30%	68.20%
CBAM (21,22)(24,25)	84.60%	88.20%	86.36%	91.80%	67.50%
CBAM (18,19)(21,22)(24,25)	87.70%	82.30%	84.91%	90.60%	67.10%

TABLE III. ABLATION EXPERIMENT RESULTS

MSEDF	GLSA	CBAM	Precision	Recall	F1	mAP@0.5	mAP@0.5:0.95
			82.0%	86.9%	84.3%	89.9%	66.1%
✓			84.3%	87.2%	85.7%	90.8%	67.4%
	✓		82.9%	86.3%	84.6%	90.3%	66.7%
		✓	87.3%	85.5%	86.4%	90.2%	66.7%
✓	✓		86.8%	85.7%	86.2%	91.5%	68.7%
✓		✓	86.2%	86.5%	86.4%	91.2%	68.1%
	✓	✓	85.0%	87.1%	86.0%	90.8%	67.0%
✓	✓	✓	**88.1%**	**87.9%**	**88.0%**	**92.5%**	**69.6%**

TABLE IV. COMPARISON OF EXPERIMENTAL RESULTS OF DIFFERENT ALGORITHMS (AVERAGE ± STD OVER 3 RUNS)

Model	Params (M)	FPS	Precision (%)	Recall (%)	F1-Score (%)	mAP@0.5 (%)	mAP@0.5:0.95 (%)
YOLOv3 (tiny)	8.7	185	85.4 ± 0.6	82.8 ± 0.5	84.1 ± 0.5	88.3 ± 0.4	60.1 ± 0.5
YOLOv5n	**1.9**	**230**	84.7 ± 0.4	86.1 ± 0.4	85.4 ± 0.4	87.5 ± 0.3	61.4 ± 0.4
YOLOv6n	4.3	205	81.3 ± 0.5	82.7 ± 0.5	82.0 ± 0.5	87.5 ± 0.4	64.5 ± 0.4
YOLOv8n (Baseline)	3.2	210	82.0 ± 0.4	86.9 ± 0.3	84.4 ± 0.4	89.9 ± 0.3	66.1 ± 0.4
YOLOv9t	7.1	175	82.7 ± 0.4	84.7 ± 0.3	83.7 ± 0.3	89.5 ± 0.2	65.9 ± 0.3
YOLOv10n	2.3	225	83.0 ± 0.3	83.2 ± 0.4	83.1 ± 0.3	88.3 ± 0.3	64.9 ± 0.3
YOLOv11n	2.6	215	81.7 ± 0.4	85.2 ± 0.3	83.4 ± 0.3	89.1 ± 0.2	65.0 ± 0.4
YOLOv12n	2.8	200	83.1 ± 0.3	86.2 ± 0.3	84.6 ± 0.3	89.2 ± 0.2	65.5 ± 0.3
RT-DETR-R18	20.0	120	86.6 ± 0.3	84.0 ± 0.4	85.3 ± 0.3	81.7 ± 0.5	59.0 ± 0.5
Mamba-YOLO	5.0	145	85.8 ± 0.3	86.4 ± 0.3	86.1 ± 0.3	90.7 ± 0.2	66.8 ± 0.3
Hyper-YOLO	6.5	135	86.6 ± 0.3	85.1 ± 0.2	85.9 ± 0.2	91.1 ± 0.2	66.8 ± 0.2
MGC-YOLOv8 (Ours)	3.95	165	**88.1 ± 0.2**	**87.9 ± 0.2**	**88.0 ± 0.1**	**92.5 ± 0.1**	**69.6 ± 0.2**

D. Visual Analysis

In object detection, visualizing results provides an intuitive means to evaluate model performance and accuracy. It offers quick insights into false positives, false negatives, and a model's ability to distinguish similar objects, thereby guiding further refinement. This section presents comparative visual results of the original and the enhanced MGC-YOLOv8 model under multi-condition industrial tool inspection scenarios, highlighting improvements in suppressing background noise, reinforcing edge features, and increasing detection confidence.

As shown in Fig. 9, baseline YOLOv8 [Fig. 9(a1)] misclassifies a background item as a chipped blade, while MGC-YOLOv8 [Fig. 9(a2)] correctly recognizes the tool and shows higher confidence, demonstrating improved resistance to background interference. In Fig. 9(b1), the baseline misses a broken blade in a cluttered environment, but the enhanced model [Fig. 9(b2)] detects it accurately, eliminating the omission. Furthermore, where YOLOv8 generates duplicate and incorrect labels [Fig. 9(c1)], MGC-YOLOv8 [Fig. 9(c2)] corrects these errors and raises confidence by 0.26 for the correct defect, indicating that the structural refinements more effectively integrate global and local features and improve discrimination between visually similar classes.

E. Confusion Matrix Analysis

The confusion matrix is an essential tool for evaluating the classification performance and robustness of object detection models. As illustrated in the normalized confusion matrices (see Fig. 10), the proposed MGC-YOLOv8 demonstrates substantial improvements over the baseline YOLOv8 across all key categories. Specifically, the detection accuracy for the Chipped Blade class increased from 0.80 to 0.84, while the Broken Blade class maintained a consistently high accuracy of 0.97, indicating strong robustness in identifying severe structural damage. Furthermore, the accuracy for the Worn Blade class improved from 0.81 to 0.86, and the Normal class increased from 0.74 to 0.83. This demonstrates the enhanced model's superior capability in recognizing fine-grained tool surface conditions.

Regarding class confusion and boundary errors, the improved architecture effectively addresses the baseline's limitations. Before the improvement, the probability of a Chipped Blade being misclassified as a Broken Blade was 0.14. This error rate dropped to 0.09 in MGC-YOLOv8, proving that the Multi-Scale Edge-Dual Fusion (MSEDF) mechanism significantly strengthens the extraction of geometric edge features, effectively distinguishing local minor defects from overall

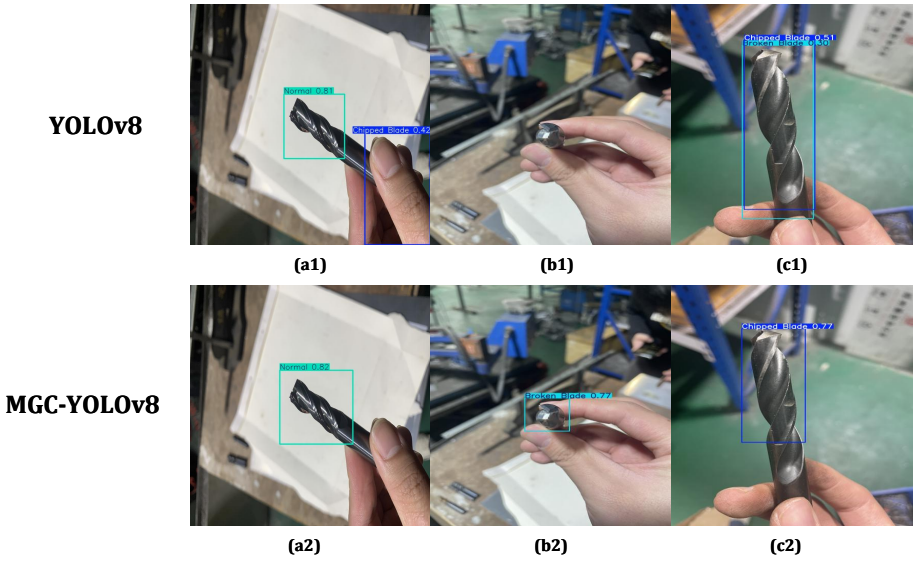
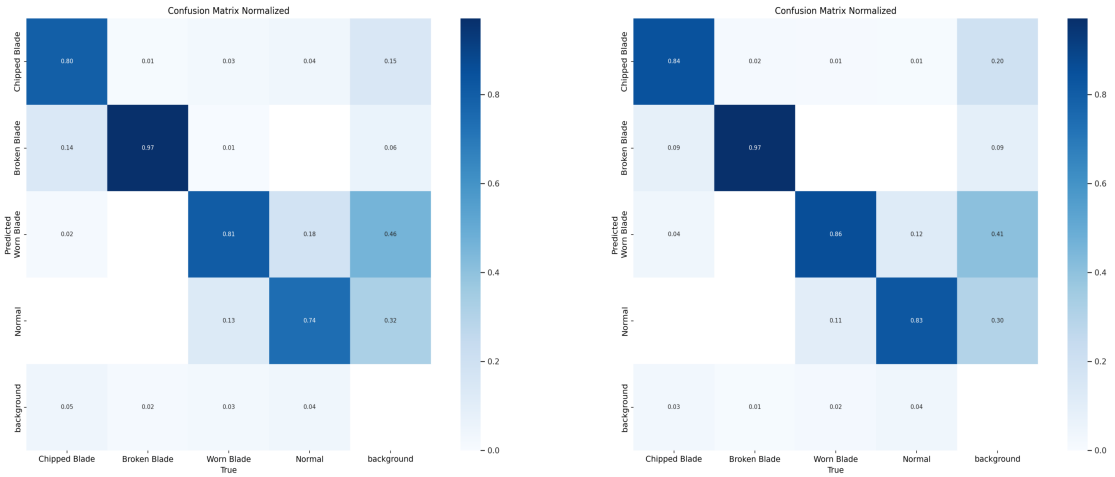


Fig. 9. Detection results before and after algorithm improvement.



(a) YOLOv8 (b) MGC-YOLOv8

Fig. 10. Confusion matrix of YOLOv8n and MGC-YOLOv8.

fractures. Similarly, the probability of a Normal surface being misjudged as a Worn Blade decreased from 0.18 to 0.12. This reduction highlights that the introduced GLSA module successfully correlates global and local semantics, capturing microscopic gradient information of wear textures to clarify ambiguous boundary errors between healthy and slightly worn surfaces.

In summary, the synergy of these proposed modules is intuitively reflected in the confusion matrix through higher diagonal accuracy and reduced inter-class confusion, validating the overall effectiveness of MGC-YOLOv8 in complex industrial tool anomaly detection.

VI. Conclusion

This study tackles the critical challenges of industrial tool anomaly detection by proposing the MGC-YOLOv8 model. The integration of a MSEDF module, a GLSA module, and a CBAM enables efficient extraction and fusion of micro-defect cues—such as cracks, wear, chipping, and breakage—on cutting-tool surfaces, yielding marked improvements in both detection accuracy and robustness. A high-quality dataset, collected under realistic factory conditions, underpins model training and validation, substantially enhancing the real-world applicability of our approach. Experimental evaluations confirm that MGC-YOLOv8 surpasses conventional detection techniques and state-of-the-art deep-learning models across

precision, recall, and mean average precision metrics.

Looking ahead, our future research will focus on several key areas to further enhance the system's robustness and applicability. To address harsh working conditions such as extremely low contrast, strong noise, or drastic changes in illumination, we plan to explore more robust feature enhancement and adaptive normalization strategies, ensuring the model can stably capture minute anomalies in diverse environments. Furthermore, although the current model demonstrates strong practical deployment potential in terms of theoretical parameter counts and inference speed, it has not yet been rigorously tested on physical embedded hardware. Therefore, a critical next step involves combining lightweight technologies—such as network pruning, knowledge distillation, and model quantization—to deploy the improved YOLOv8 on edge computing and embedded devices for genuine real-time online monitoring. Additionally, to mitigate the challenges of sample scarcity and category imbalance, we aim to introduce semi-supervised learning, cross-domain transfer learning, and diversified data augmentation methods, which will enhance the model's generalization ability across limited sample sizes and novel scenarios. Finally, integrating anomaly detection with predictive maintenance, multi-task learning, or cross-industry collaborative data training is anticipated to not only improve detection accuracy but also provide early warning support for tool life estimation and equipment health management. Through these endeavors, we aim to develop a lighter, more efficient, intelligent, and portable industrial tool monitoring system, thereby promoting the broader implementation of intelligent manufacturing across a wider range of practical applications.

ACKNOWLEDGMENT

This work was partially funded by the Henan Science and Technology Research Program under Grant 242102211009, 252102210173, the Key Research Projects of Higher Education Institutions in Henan Province under Grant 23A520022, 24A520049 and 25A520051, and the Henan Province Higher Education Institutions Young Backbone Teachers Training Program Grants 2021GGJS159.

REFERENCES

[1] F. Tao, Q. Qi, A. Liu, and A. Kusiak, "Data-driven smart manufacturing," *Journal of Manufacturing Systems*, vol. 48, pp. 157–169, 2018.

[2] J. A. Ghani, M. Rizal, M. Z. Nuawi, M. J. Ghazali, and C. H. C. Haron, "Monitoring online cutting tool wear using low-cost technique and user-friendly GUI," *Wear*, vol. 271, no. 9, pp. 2619–2624, 2011.

[3] G. Byrne, D. Dornfeld, and I. Inasaki, "Tool condition monitoring (TCM)—The status of research and industrial application," *CIRP Annals - Manufacturing Technology*, vol. 44, no. 2, pp. 541–567, 1995.

[4] X. Sun, L. Sun, and E. Wang, "Study on joint surface parameter identification method of shaft–toolholder and toolholder–tool for vertical CNC milling machine," *Machine Tool & Hydraulics*.

[5] M. P. Gómez, A. M. Hey, J. E. Ruzzante, and C. E. D'Attellis, "Tool wear evaluation in drilling by acoustic emission," *Physics Procedia*, vol. 3, no. 1, pp. 819–825, 2010.

[6] X. Shi, X. Wang, L. Jiao, Z. Wang, G. Pei, and S. Gao, "A real-time tool failure monitoring system based on cutting force analysis," *The International Journal of Advanced Manufacturing Technology*, vol. 95, pp. 5–8, 2018.

[7] M. Nouri, B. K. Fussell, B. L. Ziniti, and E. Linder, "Real-time tool wear monitoring in milling using a cutting condition independent method," *International Journal of Machine Tools and Manufacture*, vol. 89, pp. 1–13, 2015.

[8] X. Cao, B. Chen, B. Yao, and S. Zhuang, "An intelligent milling tool wear monitoring methodology based on convolutional neural network with derived wavelet frames coefficient," *Applied Sciences*, vol. 9, no. 18, p. 3912, 2019.

[9] Q. Hou, J. Sun, and P. Huang, "A novel algorithm for tool wear online inspection based on machine vision," *The International Journal of Advanced Manufacturing Technology*, 2019.

[10] Y. LeCun and L. Bottou, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[11] J. Terven, D. M. Cordova-Esparza, and J. A. Romero-Gonzalez, "A comprehensive review of YOLO architectures in computer vision: From YOLOv1 to YOLOv8 and YOLO-NAS," *Machine Learning and Knowledge Extraction*, vol. 5, no. 4, pp. 1680–1716, 2023.

[12] J. Wang, G. Xu, F. Yan, J. Wang, and Z. Wang, "Defect transformer: An efficient hybrid transformer architecture for surface defect detection," *Measurement*, 2023.

[13] F. Tang, Z. Xu, Q. Huang, J. Wang, X. Hou, J. Su, and J. Liu, "DuAT: Dual-aggregation transformer network for medical image segmentation," in *Proc. Chinese Conf. Pattern Recognition and Computer Vision (PRCV)*, pp. 343–356, 2023.

[14] S. Woo, J. Park, J. Y. Lee, and I. S. Kweon, "CBAM: Convolutional block attention module," in *Proc. ECCV*, 2018.

[15] J. Wang, W. Wang, Z. Zhang, X. Lin, J. Zhao, M. Chen, and L. Luo, "YOLO-DD: Improved YOLOv5 for defect detection," *Computers, Materials & Continua*, vol. 78, no. 1, 2024.

[16] J. Chen, Q. Wu, D. Liu, and T. Xu, "Foreground-background imbalance problem in deep object detectors: A review," *IEEE*, 2020.

[17] K. Zhu, Y. S. Wong, and G. S. Hong, "Wavelet analysis of sensor signals for tool condition monitoring: A review and some new results," *International Journal of Machine Tools and Manufacture*, vol. 49, no. 7–8, pp. 537–553, 2009.

[18] C. Zhou et al., "Vibration singularity analysis for milling tool condition monitoring," *International Journal of Mechanical Sciences*, vol. 166, 2020.

[19] Z. He, T. Shi, J. Xuan, and T. Li, "Research on tool wear prediction based on temperature signals and deep learning," *Wear*, vol. 478, p. 203902, 2021.

[20] Y. Zhou et al., "A new tool wear condition monitoring method based on deep learning under small samples," *Measurement*, vol. 189, p. 110622, 2022.

[21] P. Twardowski, M. Tabaszewski, M. Wiciak-Pikula, and A. Felusiak-Czyryca, "Identification of tool wear using acoustic emission signal and machine learning methods," *Precision Engineering*, 2021.

[22] R. Guo, H. Zhang, X. Zhi, and W. Wei, "In-process detection of tool wear based on machine vision," *Journal of Lanzhou University of Technology*, vol. 50, no. 6, pp. 33–41, 2024.

[23] J. Yu, X. Cheng, L. Lu, and B. Wu, "A machine vision method for measurement of machining tool wear," *Measurement*, vol. 182, p. 109683, 2021.

[24] Y. Tang et al., "Deep transfer hybrid model–based tool wear condition monitoring method," *Machine Tools & Hydraulics*, vol. 52, no. 20, pp. 63–71, 2024.

[25] J. Redmon et al., "You only look once: Unified, real-time object detection," in *Proc. CVPR*, 2016.

[26] T. Y. Lin et al., "Feature pyramid networks for object detection," in *Proc. CVPR*, 2017.

[27] S. Liu et al., "Path aggregation network for instance segmentation," in *Proc. CVPR*, 2018.

[28] G. Cheng et al., "Towards large-scale small object detection: Survey and benchmarks," *IEEE TPAMI*, vol. 45, no. 11, pp. 13467–13488, 2023.

[29] Y. Cui et al., "Focal network for image restoration," in *Proc. ICCV*, pp. 13001–13011, 2023.

[30] M. Yu et al., "Dense detection algorithm for ceramic tile defects based on improved YOLOv8," *Journal of Intelligent Manufacturing*, 2024.

[31] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proc. CVPR*, pp. 7263–7271, 2017.

[32] C. Li et al., "YOLOv6: A single-stage object detection framework for industrial applications," *arXiv preprint arXiv:2209.02976*, 2022.

[33] C.-Y. Wang, I.-H. Yeh, and H.-Y. M. Liao, "YOLOv9: Learning what you want to learn using programmable gradient information," in *Proc. ECCV*, 2024.

[34] A. Wang et al., "YOLOv10: Real-time end-to-end object detection," *arXiv preprint arXiv:2405.14458*, 2024.

[35] Y. Tian, Q. Ye, and D. Doermann, "YOLOv12: Attention-centric real-time object detectors," *arXiv preprint arXiv:2502.12524*, 2025.

[36] N. Carion et al., "End-to-end object detection with transformers," in *Proc. ECCV*, pp. 213–229, 2020.

[37] Z. Wang et al., "Mamba-YOLO: A simple baseline for object detection with state space model," in *Proc. AAAI*, vol. 39, no. 8, pp. 8205–8213, 2025.

[38] Y. Feng et al., "Hyper-YOLO: When visual object detection meets hypergraph computation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.