Scientific Journal of Informatics
Vol. 13, No. 2, May 2026

p-ISSN 2407-7658 http://journal.unnes.ac.id/nju/index.php/sji e-ISSN 2460-0040

Contrast-Limited Adaptive Histogram Equalization for Enhancing YOLOv8-Based Industrial Bolt Defect Detection

Muhammad Nurbaitullah[1]*, Abdul Syukur[2], Ahmad Zainul Fanani[3]
[1,2,3]Information Technology Department, Faculty of Computer Science, Universitas Dian Nuswantoro Semarang, Indonesia

Abstract.
Purpose: Defect detection in industrial bolts is crucial for ensuring product reliability, production safety, and consistent quality control in modern industrial environments. However, visual inspection of metal bolts remains challenging due to low contrast, uneven lighting, and reflective surfaces that often hide subtle defect patterns and reduce detection accuracy. Most existing YOLO-based approaches focus on architectural modifications to improve performance, which may increase model complexity and limit real-time applicability.
Methods: This study integrates Contrast-Limited Adaptive Histogram Equalization (CLAHE) with YOLOv8 to improve defect visibility prior to detection. CLAHE enhances local contrast by redistributing pixel intensities while suppressing noise amplification, thereby strengthening feature representation for deep learning-based detection. Experiments were conducted on a publicly available industrial bolt dataset annotated via Roboflow, using a 3-fold cross-validation strategy. Performance was assessed with Precision, Recall, mAP@50, mAP@50–95, FPS, and FLOPs to evaluate accuracy and real-time feasibility.
Result: Experimental results based on a 3-fold cross-validation scheme indicate that the proposed CLAHE–YOLOv8 model achieves consistent performance improvements over the baseline YOLOv8 configuration. The method obtains an average Precision of 0.9495±0.0068, Recall of 0.9028±0.0235, mAP@50 of 0.9364±0.0156, and mAP@50–95 of 0.7121±0.0037, while maintaining real-time inference performance at 29.79 FPS. These results demonstrate that contrast-based preprocessing contributes positively to detection stability and localization consistency without increasing model complexity.
Novelty: The novelty of this research lies in demonstrating that data-level contrast enhancement using CLAHE effectively improve industrial bolt defect detection performance without architectural modification, offering a practical and computationally efficient solution for real-time industrial inspection systems.

Keywords: Industrial defect detection, Bolt inspection, Deep learning, YOLOv8, CLAHE, Real-time vision
Received January 2026 / **Revised** February 2026 / **Accepted** March 2026

INTRODUCTION

Industrial automated inspection systems play a crucial role in maintaining product quality, operational safety, and production efficiency in modern manufacturing environments [1]. Mechanical fastening components, such as industrial bolts, are extensively used in automotive assemblies, heavy machinery, and structural applications, where undetected defects may result in mechanical failure, safety hazards, and economic losses [2]. Common bolt defects include surface cracks, thread deformation, wear, and localized structural irregularities, all of which require accurate and reliable detection under real industrial conditions [3].

Recent advances in deep learning have significantly improved automated visual inspection systems [4]. Among various object detection frameworks, the You Only Look Once (YOLO) family has been widely adopted in industrial inspection tasks due to its ability to achieve high detection accuracy while maintaining real-time inference capability [5]. YOLO-based detectors have demonstrated strong performance in detecting surface defects and industrial components [6]. However, when applied to metallic bolt inspection, their performance can degrade due to uneven illumination, reflective surfaces, and low contrast between defect regions and the surrounding background, which weaken feature representation and complicate accurate localization [7].

State-of-the-art research in industrial defect detection has predominantly focused on architectural optimization strategies. Various enhanced YOLO variants have been proposed, incorporating attention mechanisms, feature pyramid refinements, domain adaptation modules, and lightweight backbone

[1]*Corresponding author.
Email addresses: m.nurbaitullah@gmail.com (Bait), abah.syukur01@dsn.dinus.ac.id (Abdul), a.zainul.fanani@dsn.dinus.ac.id (Fanani)
DOI: 10.15294/sji.v13i2.41185

modifications to improve small-object detection and robustness under complex backgrounds [8]–[14], [23], [26]. These approaches have demonstrated competitive performance in detecting surface defects across different industrial domains. However, architectural enhancements often increase model depth, introduce additional computational modules, or raise overall training and inference complexity, which may limit real-time deployment in resource-constrained industrial systems [11], [24].

In practical industrial inspection scenarios, however, performance degradation frequently arises not only from architectural limitations but also from image quality challenges. Metallic bolt surfaces commonly exhibit low contrast, uneven illumination, reflective glare, and subtle defect patterns that are difficult to distinguish from the surrounding texture [7], [13]. These visual characteristics may weaken early-stage feature extraction in convolutional networks, leading to missed detections or unstable localization results. Consequently, improving image quality prior to detection represents an alternative and potentially efficient strategy for enhancing detection robustness without modifying the underlying architecture [12], [14].

Contrast-Limited Adaptive Histogram Equalization (CLAHE) is a well-established image enhancement technique designed to improve local contrast while preventing excessive noise amplification [15]. By adaptively redistributing pixel intensities values within localized regions, CLAHE enhances subtle texture variations and structural details that are critical for identifying small defects on metallic surfaces [16]. Previous studies have shown that CLAHE can improve visual clarity and learning stability in various imaging applications, including medical and industrial inspection tasks [4], [17]. Nevertheless, most YOLO-based industrial defect detection studies emphasize network-level modification, while systematic evaluation of contrast-based preprocessing as a primary performance optimization strategy remains limited.

This observation highlights a research gap: although architectural enhancement dominates state-of-the-art defect detection research, the potential of data-level contrast optimization as a competitive alternative has not been thoroughly investigated in industrial bolt inspection contexts. Therefore, this study proposes integrating CLAHE as a preprocessing strategy with YOLOv8 to enhance industrial bolt defect detection performance under low-contrast and reflective surface conditions. The objectives of this research are to (1) evaluate the impact of CLAHE on detection accuracy and localization robustness, (2) compare the proposed approach with baseline and state-of-the-art YOLO-based detectors, and (3) assess the computational efficiency and real-time feasibility for practical industrial deployment.

METHODS

This section describes the overall research workflow, dataset preparation, preprocessing strategy, model configuration, and evaluation methodology in a structured and sequential manner.

Proposed Detection Framework

The proposed defect detection framework integrates CLAHE-based preprocessing with a YOLO-based single-stage object detection architecture, as illustrated in Figure 1. The overall pipeline consists of three main stages: image preprocessing, model training, and performance evaluation.

YOLO was selected due to its proven ability to achieve high detection accuracy while maintaining real-time inference performance in industrial applications [5]. The YOLO architecture consists of three main components: a backbone network for hierarchical feature extraction, a neck module for multi-scale feature aggregation, and a detection head for predicting object locations and class probabilities [6].

Multi-scale feature fusion enables robust detection of both large bolt structures and small defect regions that frequently appear on metallic surfaces. This capability is essential for industrial bolt inspection, where defect sizes and shapes vary significantly.

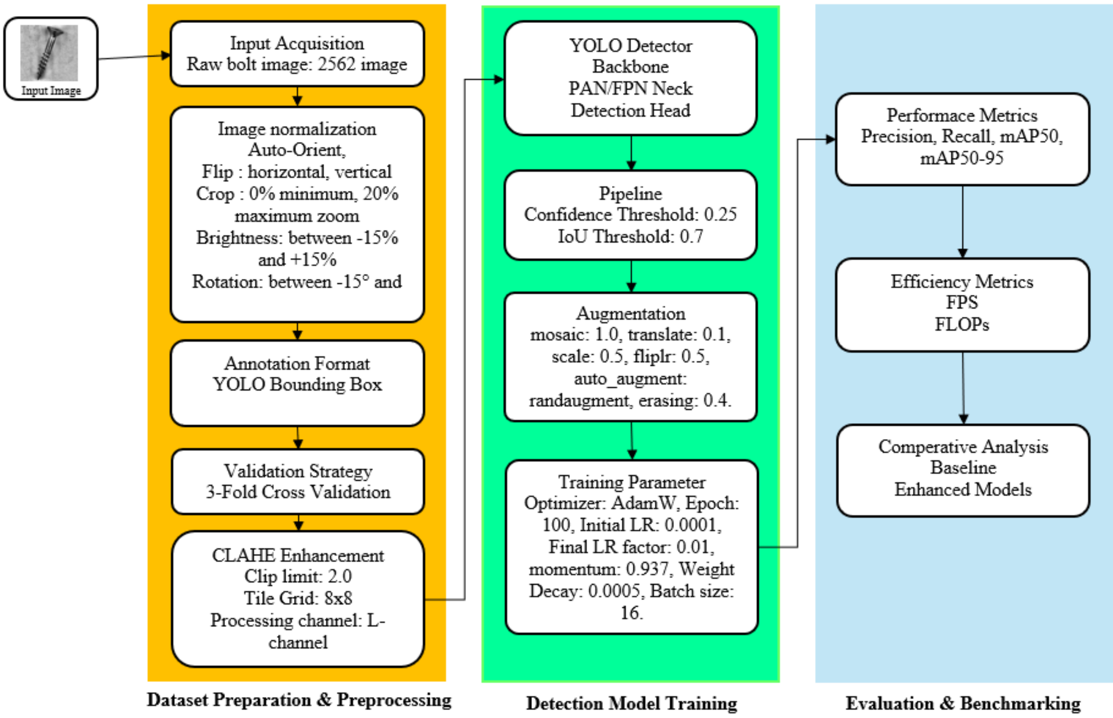


Figure 1. Overall workflow of the proposed CLAHE–YOLOv8 detection framework

Dataset and Annotation

The dataset used in this study was obtained from a publicly available industrial bolt image dataset provided on the Kaggle platform [20]. The dataset contains real-world bolt images captured under various industrial conditions, including different surface textures, orientations, and illumination variations. A total of 2562 images were used, representing diverse bolt appearances and defect characteristics such as surface scratches, pits, cracks, and thread deformation.

To support object detection tasks, all images were manually annotated using the Roboflow platform. Bounding box annotations were created for two object classes, namely bolt and defect, enabling precise localization of defect regions on the bolt surface. Roboflow was utilized to ensure annotation consistency, dataset version control, and standardized preprocessing prior to model training [21].

To improve evaluation reliability and reduce dataset bias, a 3-fold cross-validation strategy was employed [22]. In this strategy, the dataset was divided into three equally sized subsets. In each fold, two subsets were used for training and one subset was used for validation, ensuring that each image was evaluated exactly once during the validation phase.

To ensure statistical consistency and prevent single-split bias, all reported performance metrics in this study represent the mean and standard deviation obtained from the 3-fold cross-validation scheme. No independent hold-out split was used for final performance reporting.

The 3-fold cross-validation splits were constructed at the image level, ensuring that no augmented or derived versions of the same image were distributed across different folds. This strategy prevents data leakage and guarantees independent validation in each fold. Bolts dataset specification is shown in Table 1.

The statistical analysis reveals that each image contains exactly one bolt instance, confirming dataset consistency for object-level detection. Defect annotations appear as localized surface anomalies, with a total of 714 defect instances across 2562 images, resulting in an average of 1.28 bounding boxes per image as shown in Table 2.

Table 1. Bolts dataset specification

Image	Information
Format	JPEG
Resolution	512 x 512
File size	1 MB – 2 MB
Rotation	-15° and +15°
Brightness	-20% and +20%
Flip	Horizontal & Vertical
K-Fold	3-Fold cross validation
Processing Channel	L-Channel
Clip limit	2.0
Tile Grid	8x8
Total images	2562 images

Table 2. Dataset statistical summary

Item	Value
Total Images	2562
Total Bolt Bounding Boxes	2562
Total Defect Bounding Boxes	714
Average Boxes per Image	1.28
Small Defects (<96 px)	(65.13%)
Large Defects (>96 px)	(34.87%)

The defect size distribution indicates that the majority of defects (65.13%) belong to the small-size category (<96 pixels), while 34.87% are categorized as large defects (>96 pixels). This distribution suggests that the detection challenge primarily lies in capturing small-scale surface irregularities under varying illumination and reflective conditions.

The presence of fewer defect instances compared to bolt objects reflects a realistic industrial inspection scenario, where defects represent a minority yet critical category requiring reliable recall performance.

Defect annotations were defined as visible structural irregularities on the bolt surface, including cracks, scratches, surface pits, thread deformation, and material inconsistencies. Regions without observable damage were categorized as background. The annotation process ensured that each bolt instance was labeled once, while defect bounding boxes were assigned only to visually distinguishable surface anomalies.

Image Preprocessing and CLAHE Enhancement

Prior to model training, all input images were standardized through a preprocessing pipeline to ensure consistent input quality and computational efficiency. Each image was resized to a fixed resolution of 512 × 512 pixels and normalized to stabilize the learning process across all experimental configurations. Basic geometric and photometric augmentations, including horizontal and vertical flipping, rotation, scaling, and controlled brightness variation, were applied during training to improve robustness against real industrial imaging variations [11].

To specifically address low contrast and uneven illumination conditions commonly observed in metallic bolt surfaces, Contrast-Limited Adaptive Histogram Equalization (CLAHE) was incorporated as a contrast enhancement preprocessing stage. Instead of applying global histogram equalization, CLAHE operates on localized regions, enabling adaptive contrast improvement while preventing excessive noise amplification [15].

The CLAHE enhancement process employed in this study follows a structured multi-stage pipeline, as illustrated in Fig. 1. Initially, input RGB image is converted into the LAB color space, where luminance and chromatic information are separated. Contrast enhancement is then applied exclusively to the L-channel (lightness component), ensuring that color consistency is preserved while improving intensity contrast. The enhanced L-channel is subsequently recombined with the original A and B channels and transformed back into the RGB color space to produce the final enhanced image.

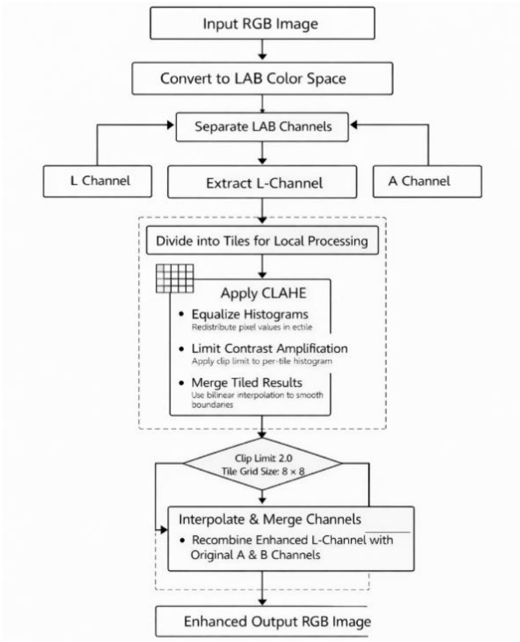


Figure 2. Flow process of CLAHE-based image enhancement

The CLAHE parameters were empirically selected based on prior studies and preliminary experimentation, with a clip limit of 2.0 and a tile grid size of 8 × 8, balancing contrast enhancement and noise suppression [16]. This structured preprocessing pipeline aims to improve the visibility of subtle defect patterns while maintaining the natural appearance of the bolt surface, thereby supporting more discriminative feature extraction during the detection stage.

Baseline and Comparator Definition
To ensure fair and reproducible benchmarking, all comparator models were explicitly defined and implemented under identical experimental settings. The YOLOv8-s and YOLOv11-s baseline architectures were obtained directly from the official Ultralytics repository (version 8.3.70).

The FSNB–YOLOv8 model incorporates architectural modifications in the feature fusion stage as described in the referenced study. The implementation was reproduced according to the architectural specification provided in the original paper, ensuring structural consistency with the proposed FSNB design.

For all models, identical hyperparameters were applied, including input resolution (512 × 512), optimizer (AdamW), initial learning rate, momentum, weight decay, batch size (16), training epochs (100), early stopping criteria (patience = 20), augmentation strategies, confidence threshold (0.25), and Non-Maximum Suppression (IoU threshold = 0.7).

CLAHE preprocessing was consistently applied prior to model input only for the specified variants, without modifying network architecture. Therefore, any performance differences observed in this study originate from architectural design or preprocessing strategy rather than discrepancies in training configuration.

Loss Functions and Optimization Strategy
Bounding box regression was optimized using the Complete Intersection over Union (CIoU) loss, which considers not only the overlap area between predicted and ground truth bounding boxes but also the distance between their center points and aspect ratio consistency [23]. The CIoU loss is formulated as shown in Equation 1:

$$\mathrm{LCIoU} = 1 - IoU + \frac{\rho 2\left(b, b^{gt}\right)}{c2} + \alpha v \tag{1}$$

where $\rho^2(\mathbf{b}, \mathbf{b}^{gt})$represents the squared Euclidean distance between the centers of the predicted and ground truth bounding boxes, $c$denotes the diagonal length of the smallest enclosing box, and $v$measures aspect ratio consistency.

The weighting factor $\alpha$is defined as shown in Equation 2:

$$\alpha = \frac{v}{(1-IoU)+v} \tag{2}$$

This formulation improves localization stability and prevents unstable bounding box estimation, particularly for small defect regions.

For classification, Binary Cross Entropy (BCE) loss was employed to distinguish between defective and non-defective regions. Equation 3 is the BCE losses definition.

$$\mathcal{L}_{BCE} = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log(p_i) + (1 - y_i)\log(1 - p_i)] \tag{3}$$

where y_i denotes the ground truth label and p_i represents the predicted probability.

Training Configuration

All models were trained using a unified experimental configuration to ensure fair comparison and reproducibility [25]. Training was conducted using the Ultralytics framework for 100 epochs, with early stopping applied using a patience value of 20 epochs. A batch size of 16 was selected to balance training stability and GPU memory usage.

The AdamW optimizer was employed with an initial learning rate of 0.0001, momentum of 0.937, and weight decay of 0.0005 [26]. Data augmentation techniques, including random scaling, translation, horizontal flipping, and brightness variation, were applied during training to improve generalization under realistic industrial conditions [11].

Table 3. Training configuration

Configuration Aspect	Technical Spesification
Input Resolution	512 × 512
Batch Size	16
Epochs	100
Early Stopping	Patience = 20
Optimizer	AdamW
Initial Learning Rate	0.0001
Learning Rate Decay Factor	0.01
Momentum	0.937
Weight Decay	0.0005
Augmentation	Mosaic 1.0, Translate 0.1, Scale 0.5, FlipLR 0.5, RandAugment Enabled, Erasing 0.4
Confidence Threshold	0.25
IoU Threshold (NMS)	0.7
Validation Strategy	3-Fold Cross Validation
CLAHE Parameters	Clip Limit: 2.0, Grid Size: 8×8, Channel: L (LAB)

All performance metrics reported in this study represent the mean and standard deviation obtained from a 3-fold cross-validation scheme. No single-split results are used for final performance reporting to ensure statistical reliability and evaluation consistency.

All comparative models presented in Table 3 were re-implemented and trained under identical experimental settings using the same dataset, augmentation strategy, optimizer configuration, and cross-validation scheme to ensure fair benchmarking.

Reproducibility and Experimental Environment

All experiments were conducted using Google Colab with NVIDIA Tesla T4 GPU acceleration. The implementation was developed in Python using the Ultralytics framework (version 8.3.70) with PyTorch backend. The dataset was obtained from a publicly available Kaggle repository and exported in YOLO format for training and evaluation. No additional resizing, compression, or external preprocessing beyond the described CLAHE pipeline and augmentation configuration were applied. Random seeds were fixed across experiments to ensure consistent initialization and reproducibility of results. All cross-validation experiments were executed under identical runtime conditions. The complete training configuration is summarized in Table 2 to facilitate experimental replication and ensure transparency of the benchmarking process.

Performance Evaluation Metrics

Performance evaluation of the proposed model was conducted using standard quantitative metrics commonly applied in industrial defect detection research, particularly for YOLO-based deep learning systems [3], [6], [11]. The selected metrics include Intersection-over-Union (IoU), Precision, Recall, and mean Average Precision (mAP) at IoU thresholds of 0.50 and 0.50–0.95. These metrics enable objective measurement of detection accuracy, localization consistency, and classification capability across all model configurations.

1) Precision

Precision reflects the proportion of correctly predicted defect instances among all detected prediction outputs. It is defined in Equation 4:

$$Precision = \frac{TP}{TP + FP} \tag{4}$$

where:

- TP= True Positives
- FP= False Positives

Higher precision signifies fewer false alarms, which is critical in automated inspection systems to prevent unnecessary product rejection [27].

2) Recall

Recall evaluates the model's ability to detect all existing true defect instances and is formulated in Equation 5.

$$Recall = \frac{TP}{TP + FN} \tag{5}$$

where:

- FN= False Negatives

High recall is essential in industrial safety-critical environments where missed defect detection may lead to mechanical failure [24], [28].

3) Average Precision (AP)

Average Precision (AP) is computed from the area under the Precision-Recall curve. The continuous integral form is expressed as Equation 6.

$$AP = \int_0^1 P(R)\, dR \tag{6}$$

where $P(R)$represents precision as a function of recall.

In practice, AP is computed using discrete recall thresholds as shown in Equation 7:

$$AP = \sum_{n=1}^{N} (R_n - R_{n-1}) \cdot P\_n \tag{7}$$

4) Mean Average Precision (mAP)

The final metric, mean Average Precision, provides the overall detection score across all classes is computed in Equation 8 [29].

$$mAP = \frac{1}{C}\sum_{i=1}^{C} AP_i \tag{8}$$

where:

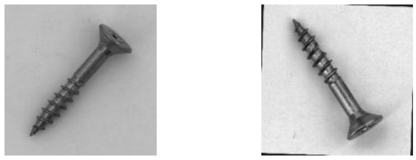
- C= number of object classes
- AP_i= Average Precision for class i

RESULT AND DISCUSSION

This section presents the experimental results and discussion based on the implementation of the CLAHE–YOLOv8 method described in the Methods section. Evaluations were conducted to measure the improvement in bolt defect detection performance resulting from the application of CLAHE preprocessing, and to compare it with the baseline model and other YOLO variants. All experiments were conducted using the same industrial bolt dataset, identical training configurations, and a 3-fold cross validation scheme to ensure fairness of comparison.

Visual Impact of CLAHE Enhancement

To understand the contribution of CLAHE visually, a qualitative analysis was performed on the images before and after CLAHE preprocessing in Figure 3.



(a) Original image (b) CLAHE image

Figure 3. CLAHE-enhanced image

Visually, CLAHE images show a significant increase in local contrast, especially in areas with smooth textures and small defects. The boundaries between defective areas and normal surfaces become clearer, without causing excessive artifacts or noise. This increased visibility is particularly important for convolutional networks in the early stages, as low-contrast features that were previously difficult to recognize become easier to extract.

These visual findings are consistent with the increase in Recall and mAP@50–95 values, indicating that the model is capable of detecting defects that were previously difficult to identify due to low contrast.

Qualitative Result

Figure 4 presents qualitative detection results obtained by the proposed detection framework on representative bolt images from the test dataset. The visualized results illustrate both object-level localization of the bolt and fine-grained detection of defect regions under various orientations, surface textures, and illumination conditions. The blue bounding boxes represent successful bolt localization, while the cyan bounding boxes indicate detected defect regions.

From the visual results, it can be observed that the detector consistently identifies the bolt object with a single, well-aligned bounding box across all samples. This behavior confirms the stability of the backbone and multi-scale feature aggregation in capturing the global structure of the bolt, regardless of rotation, scale variation, or background texture. The absence of fragmented or duplicated bolt bounding boxes indicates effective Non-Maximum Suppression (NMS) behavior and robust object confidence estimation, which aligns with the high Precision values reported in the quantitative evaluation.

More importantly, the defect detection results demonstrate the model's ability to localize small and visually subtle surface anomalies along the bolt shaft and thread region. The detected defect bounding boxes are spatially compact and accurately aligned with regions exhibiting surface irregularities, such as scratches, pits, or deformation marks. These defects are often characterized by weak contrast and minimal geometric variation relative to the surrounding metallic surface, making them challenging to detect using standard object detectors. The successful localization of such regions provides qualitative confirmation of the improved feature separability achieved through contrast enhancement and optimized feature extraction.

Several examples in Fig. 4 show multiple defect detections on a single bolt instance, indicating that the model does not assume a fixed number of defects per object but instead evaluates local surface regions independently. This behavior is particularly relevant for industrial inspection scenarios, where a single product may contain multiple defect types or defect occurrences. The ability to detect multiple defect regions without excessive false positives supports the robustness of the classification head and confidence calibration, consistent with the high Recall values observed in the quantitative results.

The qualitative results also reveal that the detector maintains detection reliability across different bolt orientations and background conditions. Defects are successfully detected even when the bolt is rotated at oblique angles or partially occluded by background texture variation. This observation suggests that the learned features are not overly sensitive to spatial orientation and that the model generalizes well across geometric transformations introduced during training through data augmentation.

Overall, the qualitative detection results shown in Fig. 4 provide strong visual evidence that the proposed framework is capable of accurately localizing both bolt objects and fine-grained defect regions under realistic industrial conditions. These observations corroborate the quantitative performance gains reported earlier and demonstrate that the proposed method delivers not only numerical improvements but also reliable and interpretable detection behavior suitable for real-world industrial bolt inspection applications.

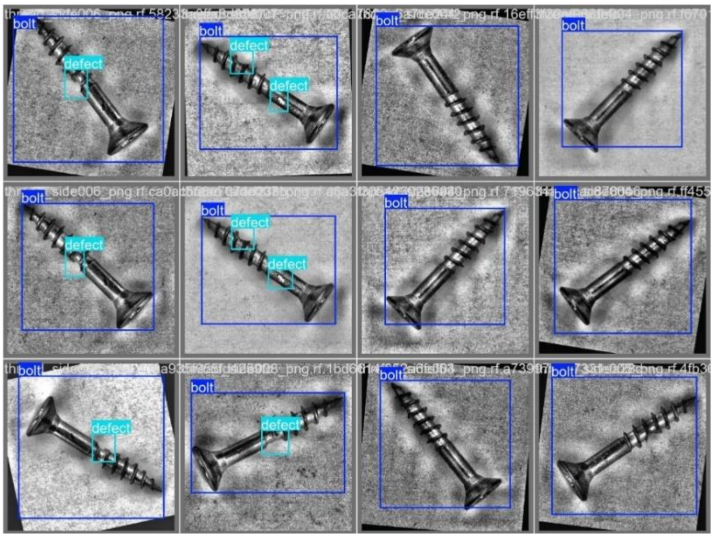


Figure 4. Detection result

Confusion Matrix Analysis

To evaluate classification errors in more detail, the confusion matrix from the CLAHE–YOLOv8 model was used.

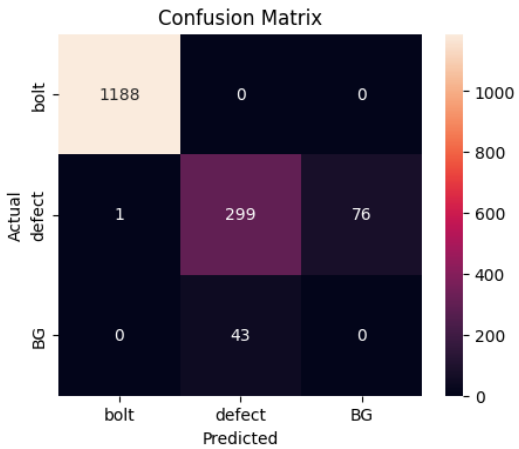


Figure 5. Confusion matrix

Based on the confusion matrix shown in Figure 5, the performance of the CLAHE–YOLOv8 model in classifying bolt, defects, and backgrounds can be analyzed in detail. In the objects class (bolt), the model showed excellent performance with 1188 instances correctly classified as objects and no misclassification into other classes. This shows that the model has a very high ability to recognize screw objects consistently and stably.

In the defect class, 299 instances were successfully detected and correctly classified as defects. However, there were 76 defect instances that were misclassified as background, indicating that some defects with very subtle visual characteristics or low contrast are still difficult to distinguish from the background. In addition, there was 1 defect instance that was misclassified as a bolt object, although this number of errors is relatively very small compared to the number of correct detections.

For the background class, there were 43 background instances that were incorrectly detected as defect. This finding indicates that under certain conditions, the surface texture of the bolt or visual noise in the background resembles the defect pattern, triggering false positive predictions in the defect class. However, no background classification errors to the objects class were found, indicating that the model is able to clearly distinguish bolt objects from the background.

Overall, this confusion matrix shows that the model has a very high accuracy rate in detecting bolt objects, as well as good sensitivity in detecting defects. The classification errors that occurred were mainly at the boundary between the defect and background classes, which is a common challenge in metal surface inspection due to small defect sizes, texture variations, and low contrast. These results are consistent with the previous Precision–Recall analysis and confirm that the application of CLAHE helps improve defect detection sensitivity without causing a significant increase in errors in other classes.

3-Fold Cross Validation Result

Model performance evaluation was conducted using a 3-fold cross validation scheme to measure the stability and consistency of the CLAHE–YOLOv8 model on different data partitions. Each fold was trained and tested independently with identical training configurations. The evaluation results for each fold are shown in Table 4.

Table 4. Performance results per fold in 3-fold cross validation

Fold	Precision	Recall	mAP50	mAP50–0.95
Fold 1	0.942117	0.875698	0.918399	0.709446
Fold 2	0.955661	0.917704	0.944766	0.716305
Fold 3	0.950605	0.914931	0.945964	0.710583

Based on the results for each fold, it can be observed that the model's performance is relatively consistent across all data divisions. The Precision value ranges from 0.94 to 0.96, while Recall ranges from 0.87 to 0.92. The mAP@50 and mAP@50–95 values also show little variation between folds, indicating that the model does not depend on a particular data division.

To provide a more comprehensive picture of the model's stability, the mean and standard deviation of the three folds were calculated, as shown in Table 5.

Table 5. Mean and standard deviation of 3-fold cross validation results

Metrix	Mean	Std
Precision	0.949461	0.006844
Recall	0.902778	0.023492
mAP@50	0.936376	0.015581
mAP@50-95	0.712111	0.003676

The mean Precision value of 0.949461 with a standard deviation of 0.006844 indicates that the model has a high and stable level of accuracy in identifying correct predictions. The average recall of 0.902778 with a standard deviation of 0.023492 indicates that the model's ability to detect all defects is relatively consistent, despite slight variations between folds due to differences in the distribution of defective samples.

The mAP@50 and mAP@50–95 values of 0.936376 ± 0.015581 and 0.712111 ± 0.003676, respectively, indicate that the bounding box localization performance remains stable at various IoU thresholds. The relatively small standard deviation, particularly in mAP@50–95, confirms that the CLAHE–YOLOv8 model has good robustness against training data variations and does not experience significant performance fluctuations between folds.

Overall, the results of this 3-fold cross validation evaluation prove that the performance improvement obtained from applying CLAHE preprocessing is consistent and reproducible, thus supporting the feasibility of the proposed method for application in real industrial bolt inspection systems.

Quantitative Detection Results

Table 6 presents the average detection performance of all models obtained from the 3-fold cross-validation scheme. The reported values represent the mean performance across folds, ensuring evaluation robustness and statistical consistency. Compared to the YOLOv8 baseline, the proposed CLAHE–YOLOv8 model demonstrates consistent improvements across all accuracy metrics, particularly in mAP@50 and mAP@50–95, indicating enhanced localization stability under varying IoU thresholds.

Table 6 presents the averaged detection performance across the 3-fold cross-validation scheme. The reported values represent mean ± standard deviation, ensuring statistical consistency and robustness evaluation. Compared to the YOLOv8 baseline, the proposed CLAHE–YOLOv8 model demonstrates consistent improvements across all evaluation metrics, particularly in mAP@50 and mAP@50–95, indicating enhanced localization stability under varying IoU thresholds.

The mAP@50 and mAP@50–95 values also show consistent improvements, indicating that CLAHE not only improves classification capabilities but also the stability of bounding box localization at various IoU thresholds. Despite the significant improvement in accuracy, the inference speed of CLAHE–YOLOv8 remains within the real-time range (29.79 FPS), thus still meeting the needs of direct industrial inspection.

Table 6. Result of comparative performance

Model	Precision	Recall	mAP50	mAP50–0.95	FPS
CLAHE–YOLOv8	**0.9495**	**0.915**	**0.946**	**0.7121**	**29.79**
YOLOv11 Baseline	0.9591	0.9365	0.934	0.8013	23.8
FSNB–YOLOv8	0.9437	0.878	0.9098	0.5945	32.1
FSNB–YOLOv8 + CLAHE	0.9447	0.8739	0.9025	0.6085	20.3
YOLOv11 + CLAHE	0.8744	0.8272	0.8666	0.6423	27.31
YOLOv8 Baseline	0.857	0.787	0.818	0.748	28.08

Comparison with Other Models and Practical Implications

Compared to recent architecture-driven approaches such as DSL-YOLO and hybrid attention-based YOLO variants, which introduce structural modifications to improve feature representation [28], the proposed CLAHE–YOLOv8 framework demonstrates that data-level contrast enhancement alone can provide competitive improvements without increasing model complexity. This finding offers an alternative perspective to the dominant trend of architectural expansion in industrial defect detection research. Since all models were trained under identical experimental settings using the same dataset, input resolution, optimizer configuration, and augmentation strategy, the reported performance differences can be directly attributed to architectural design and preprocessing mechanisms rather than inconsistencies in training conditions.Compared to architecture-based approaches such as FSNB–YOLOv8, CLAHE–YOLOv8 demonstrates superiority in maintaining a balance between accuracy and computational efficiency. Although FSNB–YOLOv8 has a higher FPS, its lower mAP@50–95 values indicate a lack of localization consistency across various IoU thresholds.

Meanwhile, YOLOv11 Baseline shows competitive performance, but with lower inference speed. Interestingly, applying CLAHE to YOLOv11 actually reduces accuracy performance, indicating that the YOLOv11 architecture already has a strong internal mechanism for handling contrast variations, so that additional preprocessing can interfere with the learned feature distribution.

From an industrial implementation perspective, CLAHE–YOLOv8 offers a more practical solution because performance improvements are achieved through optimization at the data level, without adding significant architectural complexity or computational overhead.

Computational Efficiency and Deployment Feasibility

Beyond detection accuracy, industrial deployment requires computational efficiency. As shown in Table 7, CLAHE–YOLOv8 requires 28.4 GFLOPs with 11.1 million parameters, achieving 29.79 FPS, which satisfies real-time inspection requirements. FSNB–YOLOv8, although achieving the highest FPS (32.10), requires significantly higher computational cost (71.9 GFLOPs), potentially limiting its applicability in resource-constrained environments. The YOLOv11 baseline exhibits the lowest computational complexity (21.3 GFLOPs) but achieves lower FPS compared to CLAHE–YOLOv8, indicating that architectural efficiency does not always directly translate into higher throughput. From a deployment perspective, CLAHE–YOLOv8 offers the most balanced trade-off between accuracy, robustness, and computational efficiency.

Table 7. Computational complexity analysis

Model	Parameters	FLOPs (GFLOPs)
YOLOv8 Baseline	11,126,358	28.6
CLAHE–YOLOv8	11,126,358	28.4
FSNB–YOLOv8	12,870,994	71.9
FSNB–YOLOv8 + CLAHE	12,870,994	71.9
YOLOv11 Baseline	9,413,574	21.3
YOLOv11 + CLAHE	9,413,574	21.3

Robustness and Consistency Analysis of Detection

To analyze the model's behavior towards changes in the confidence threshold, Precision–Confidence and Recall–Confidence curves are used.

Based on the Precision–Recall curve shown in Figure 6, it can be observed that the CLAHE–YOLOv8 model shows excellent detection performance in the bolt class and more challenging performance in the defect class. The curve for the bolt class is very close to the maximum precision area in almost the entire recall range, with an Average Precision (AP) value of 0.995. This indicates that bolt objects have consistent visual characteristics that are easily recognized by the model, resulting in a very low detection error rate.

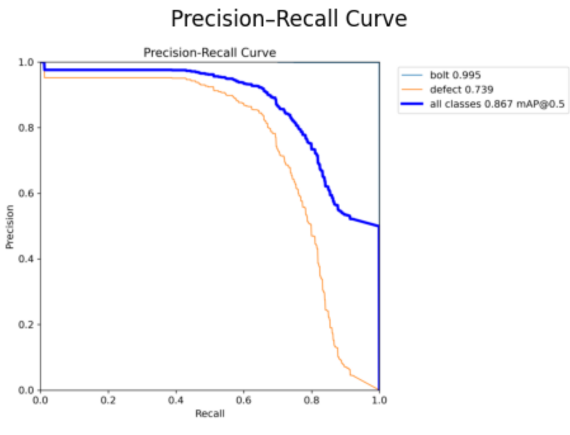


Figure 6. Precision–confidence dan recall–confidence curve

In contrast, the Precision–Recall curve for the defect class shows a faster decline in precision as recall increases, with an AP value of 0.739. This pattern indicates that defect detection has a higher level of complexity than bolt object detection, mainly due to the relatively small size of defects, low contrast, and similarity in texture between the defect area and the bolt surface. At high recall values, the model tends to increase detection sensitivity so that some predictions with low confidence are also detected, which results in a decrease in precision.

The combined curve (all classes) shows a mAP@0.5 value of 0.867, which reflects a good balance between precision and recall overall. The relatively smooth curve shape without sharp fluctuations indicates that the model has good prediction stability against changes in the confidence threshold. This indicates that the detection system does not depend on a specific threshold point, but is able to maintain consistent performance under various operational conditions.

Overall, this Precision–Recall curve analysis shows that the application of CLAHE preprocessing helps the model maintain a high recall rate without extreme precision degradation, especially for defect classes with difficult visual characteristics. This finding reinforces the results of previous quantitative evaluations and confirms that the CLAHE–YOLOv8 model has a good balance between sensitivity and detection reliability for industrial bolt inspection applications.

Limitations of the Study

Despite the promising performance achieved, this study has several limitations. First, the dataset used represents a specific industrial bolt inspection scenario and may not fully capture extreme defect variations or micro-scale anomalies. Second, no extremely small defect instances (<32 pixels) were present in the dataset, limiting evaluation on ultra-small object detection scenarios. Third, while CLAHE improves contrast under current illumination conditions, its effectiveness under extreme lighting or high-glare environments requires further investigation. Future studies should evaluate adaptive contrast enhancement strategies and cross-dataset generalization to improve robustness.

CONCLUSION

This study demonstrates that integrating Contrast-Limited Adaptive Histogram Equalization (CLAHE) as a preprocessing strategy significantly enhances YOLOv8-based industrial bolt defect detection under low-contrast and reflective surface conditions. Through 3-fold cross-validation, the proposed CLAHE–YOLOv8 framework achieves stable and consistent performance improvements without increasing model complexity or computational burden. The findings indicate that data-level optimization can serve as a viable alternative to architectural modification for improving robust detection in industrial inspection systems. From a practical perspective, the proposed approach provides a computationally efficient solution suitable for real-time deployment in manufacturing environments, particularly where hardware resources are limited. The research contributes to the growing field of industrial vision systems by highlighting the importance of preprocessing-driven optimization strategies in defect detection tasks. Future work may explore adaptive CLAHE parameter tuning, integration with lightweight attention mechanisms, and evaluation on embedded or edge computing platforms to assess real-world deployment feasibility.

REFERENCES

[1] K. Halloum et al, “Enhancing Medical Image Classification through Transfer Learning and CLAHE Optimization,” National Library of Medicine, Volume 21, Issue 1, 2025. https://doi.org/10.2174/0115734056342623241119061744

[2] I. Kuric, et al, “Approach to Automated Visual Inspection of Objects Based on Artificial Intelligence,” Applied Sciences., vol. 12(2), 864, 2022. https://doi.org/10.3390/app12020864

[3] Edmundo Casas, et al, “A comparative study of YOLOv5 and YOLOv8 for corrosion segmentation tasks in metal surfaces,” Array, Volume 22, 2024. https://doi.org/10.1016/j.array.2024.100351.

[4] F. Syauqi, et al, "Edge AI-Based Defect Detection in White Pepper (Piper Nigrum L.) Using CLAHE-Based Preprocessing and YOLO," IEEE International Conference on Networking, Intelligent Systems, and IoT (ICONS-IoT), 2025, pp. 18-23, doi: 10.1109/ICONS-IoT65216.2025.11211242.

[5] Chu, Y. et al, “A Lightweight Strip Steel Surface Defect Detection Network Based on Improved YOLOv8,” Sensors, 24, 6495, 2024. https://doi.org/10.3390/s24196495

[6] Jiajun Ren et al., “YOLOv8-WD: Deep Learning-Based Detection of Defects in in Automotive Brake Joint Laser Welds,” Applied Sciences. 2025; 15(3):1184. https://doi.org/10.3390/app15031184.

[7] O. C. Han, U. Kutbay, “Detection of Defects on Metal Surfaces Based on Deep Learning,” Applied Sciences., 15(3):1406, 2025. https://doi.org/10.3390/app15031406

[8] Guoxiang Hua et al, “An enhanced YOLOv8-based bolt detection algorithm for transmission line,” IET Generation, Transmission, & Distribution, 18, 4065–4077, 2024. https://doi.org/10.1049/gtd2.13330

[9] Dayana Palma-Ramírez et al., “Deep convolutional neural network for weld defect classification in radiographic images,” Research article, Volume 10, Issue 9, e30590, 2024.

[10] R. Khanam, et al, “Comparative Performance Evaluation of YOLOv5, YOLOv8, and YOLOv11 for Solar Panel Defect Detection,” MDPI, Vol. 5, No. 1, Article 6, 2025. https://doi.org/10.3390/solar5010006
[11] M. Shahin et al, “Improving the Concrete Crack Detection Process via a Hybrid Visual Transformer Algorithm,” Sensors, 24(10):3247, 2024. https://doi.org/10.3390/s24103247
[12] Chen Li et al, "A domain adaptation YOLOv5 model for industrial defect inspection," Measurement (Elsevier), Volume 213, 2023. https://doi.org/10.1016/j.measurement.2023.112725
[13] Jiajian Liu et al, “FDFNet: An Efficient Detection Network for Small-Size Surface Defect Based on Feature Differentiated Fusion,” Digital Signal Processing, Volume 167,, 2025. https://doi.org/10.1016/j.dsp.2025.105432
[14] G. Yao, et al, HP-YOLOv8: High-Precision Small Object Detection Algorithm for Remote Sensing Images,” Sensors, 2024, 24(15), 4858; https://doi.org/10.3390/s24154858.
[15] Ubaid Ullah et al, " IoT-enabled computer vision-based parts inspection system for SME 4.0," Microprocessors and Microsystems, Volume 87, 2021, 104354. https://doi.org/10.1016/j.micpro.2021.104354
[16] L. Wang et al, “Real-Time Steel Surface Defect Detection with Improved Multi-Scale YOLO-v5,” Processes (MDPI), vol. 11(5):1357, 2023. https://doi.org/10.3390/pr11051357
[17] Wang S. et al, Automatic Detection and Classification of Steel Surface Defect Using Deep Convolutional Neural Networks, Metals. 2021; 11(3):388. https://doi.org/10.3390/met11030388.
[18] Zhiyong Ju et al, “GLDS-YOLO: An Improved Lightweight Model for Small Object Detection in UAV Aerial Imagery,” Electronics, 14 (19), 3831, 2025. https://doi.org/10.3390/electronics14193831
[19] Avishek Mukherjee et al, “Expert-aware multi-stage AI-assisted image processing pipeline for accurate weld defect detection from radiographs: a use case for metal fabrication industries,” Nondestructive Testing and Evaluation, Vol 1–63, 2025. https://doi.org/10.1080/10589759.2025.2575884.
[20] Li, R., Zhao, L., Wei, H. et al. “Optimized YOLOv8 for lightweight and high-precision metal surface defect detection in industrial applications,” Mach Learn 114, 226. 2025. https://doi.org/10.1007/s10994-025-06857-3.
[21] Irfan Ullah Khan et al, “Deep Learning-Based Surface Defect Detection in Steel Products Using Convolutional Neural Networks,” Mathematical Modelling of Engineering Problems, 2024. https://doi.org/10.18280/mmep.111113.
[22] Anping Wan et al, " Deep learning-based intelligent visual inspection for defect detection in manufacturing," Next Generation Manufacturing Systems (Elsevier)., vol. 2, 2025.
[23] Z. Wang et al., “Research on a Metal Surface Defect Detection Algorithm Based on DSL-YOLO,” Sensors, vol. 24, no. 19, 6268, 2024.
[24] B. Tang, W. Chen, H. Zhang, “Review of surface defect inspection technology of steel products,” IET Image Processing., 2023. https://doi.org/10.1016/j.nexres.2025.100424
[25] Dongxu Bai et al, “Surface defect detection methods for industrial products with imbalanced samples: A review of progress in the 2020s,” Engineering Applications of Artificial Intelligence, vol. 130, 107697, 2024.
[26] Deng, H et al, “USD-YOLO: An Enhanced YOLO Algorithm for Small Object Detection in Unmanned Systems Perception,” Applied Sciences, 15(7), 3795, 2025. https://doi.org/10.3390/app15073795
[27] Hütten, N. et al, " Deep Learning for Automated Visual Inspection in Manufacturing and Maintenance: A Survey of Open- Access Papers," Applied System Innovation, Vol. 7(1):11, 2024. https://doi.org/10.3390/asi7010011
[28] Makaro Mao et al, “YOLO Object Detection for Real-Time Fabric Defect Inspection in the Textile Industry: A Review of YOLOv1 to YOLOv11,” Sensors, 25(7), 2270, 2025. https://doi.org/10.3390/s25072270
[29] Hou M. et al, “CNN-based defect detection in manufacturing,” Advanced Control for Applications: Engineering and Industrial Systems, 6(4):e196, 2024. https://doi.org/10.1002/adc2.196
[30] Yongchuan Lin et al, “HIP-FCA-Net: an intelligent detection method for pixel-level fine cracks on the surface of castings integrating image adaptive enhancement function,” Measurement Science and Technology, Volume 37, Number 4, 2026. DOI 10.1088/1361-6501/ae33de
[31] Y. Ma et al, “Surface defect inspection of industrial products with object detection deep networks: a systematic review,” Artificial Intelligence Review, 2024, doi:10.1007/s10462-024-10956-3.