INDIAN JOURNAL OF SCIENCE AND TECHNOLOGY

RESEARCH ARTICLE

OPEN ACCESS

Received: 28-11-2024

Accepted: 02-02-2025

Published: 25-02-2025

Citation: Khemani LP, Yadav RK, Doshi TV (2025) Novel Optimized Image Enhancement and Deep Learning Based Crack Detection Algorithm in Non-Destructive Testing. Indian Journal of Science and Technology 18(7): 487-503. https://doi.org/10.17485/IJST/v18i7.3874

* **Corresponding author**.

lucky.khemani@i2atechs.com

Funding: None

Competing Interests: None

Published By Indian Society for Education and Environment (iSee)

ISSN
Print: 0974-6846
Electronic: 0974-5645

Novel Optimized Image Enhancement and Deep Learning Based Crack Detection Algorithm in Non-Destructive Testing

Lucky Pratap Khemani[1]*, Ravindra Kumar Yadav[2], Twinkle V Doshi[3]

1 Research Scholar, KPGU & CEO, AI and Imaging, Innovate2Automate technologies Pvt Ltd, India
2 Associate Professor, Electrical Engineering, KPGU, Vadodara, Gujarat, India
3 Assistant Professor, Department of Computer Application, Faculty of Science, The M.S. University, Vadodara, Gujarat, India

Abstract

Objectives: The main objectives are to enhance image clarity for crack detection in steel pipes using a combination of denoising techniques, adaptive contrast enhancement, and YOLOv10-based detection with fine-tuned parameters in NDT. Industries such as oil and gas and chemical processing heavily rely on these techniques to ensure the integrity of their pipelines. **Methods:** The process involves reducing noise to improve image quality, applying CLAHE for localized contrast enhancement, and leveraging YOLOv10's deep learning capabilities for crack detection. Key parameters are adjusted to maximize the model's precision, recall, and confidence scores. **Findings:** The method improves the accuracy of crack detection in NDT scenarios considerably. Noise reduction and CLAHE enhance the visibility of defects while preserving critical information in images. Fine-tuning the parameters of YOLOv10 results in performance that is superior to previous methods, with a confidence rate of over 92% and precision of over 99% for detecting cracks under various conditions. This approach shows robustness against noise and illumination changes, which decreases detection errors and increases reliability in industrial inspections. Overall, it provides a robust and efficient method for accurate defect identification. **Novelty:** This work integrates an optimized denoising algorithm, optimized CLAHE algorithm, and YOLOv10 with customized optimal parameter tuning to achieve exceptional crack detection accuracy and efficiency in the context of NDT. The creation of a real-world augmented dataset is itself a significant achievement in addressing real-world problems. Achieves high performance training in just 50 epoch, making it specially designed for real-time business applications where we experience different drastic change in data capture conditions. Also Inference model with preprocessing performance is 40-50 fps with T4 GPU, which can be further enhanced by using the latest next versions of GPU.

Keywords: Deep Learning; CLAHE; YOLOv10; Image Denoising; Crack Detection; Non-Destructive Testing

1 Introduction

Non-Destructive Testing (NDT) becomes an indispensable diagnostic tool in the assessment of materials and structures by not causing any form of physical harm. The roles of imaging techniques, whether X-ray, ultrasonic, or infrared, constitute the basis of crack and flaw detection. Yet these methods and techniques still have significant challenges and often have trouble regarding accurate and reliable defect identification. This means low contrast and noise lead to poor image quality that hinders some critical features in the image, and sometimes it might be impossible to see defects on the NDT image. Certain gaps in this regard of enhancing the image quality and the automation of detection of defects remain unexplored by the state-of-the-art techniques.

The principal issue in NDT images is low-contrast images; crucial details of the defect get lost in noisier and worse lighting conditions. To overcome such drawbacks, contrast enhancement techniques became prominent, such as AHE and its optimized variant called CLAHE. CLAHE has been particularly effective in avoiding over-amplification of noise, and conserving localized contrast. CLAHE has also received considerable attention for use in the detection of defects by their image in NDT applications. However, even these methods have limitations in handling extreme noise or ensuring consistent enhancement across varying defect types and environments.

Hardware noise, which includes salt-and-pepper and Gaussian noise, greatly degrades image quality. Salt-and-pepper noise introduces random black or white spots in the image, while Gaussian noise introduces random intensity variations that make fine cracks difficult to detect. Most denoising algorithms fail to find the balance between noise reduction and preserving critical features of defects; thus, they either lead to false positives or missed detections.

Apart from image enhancement and denoising, the accuracy of defect detection systems remains of high priority. Many advanced crack detection methods depend upon deep learning algorithms, particularly YOLO (You Only Look Once), and have provided promising results with real-time object detection applications. These, however, have several challenges for crack detection regardless of the sizes, shapes, or orientations, which may appear more complicated especially when contaminants on the surface are such things as corrosion or weld marks. It also means real-time processing requirements, with diverse environmental conditions including temperature fluctuations and vibrations added to the complexity of automatic detection systems.

Research Gaps:

Despite significant developments in the field, gaps persist in research:

- Image denoising methods existing today have lacked robustness in the combination of noise types - namely, salt-and-pepper and Gaussian noise- while preserving defect integrity.
- Contrasts in NDT images may be improved by CLAHE, but it is hardly applicable in extreme noise conditions and complex surface textures.
- Current deep learning-based crack detection systems suffer from non-generalization across a range of datasets and environmental conditions and often exhibit inconsistent performance.
- Current methods require extensive training time and resources, as well as large datasets.

To fill this gap, this paper suggests a new optimal denoising algorithm that can simultaneously remove salt-and-pepper and Gaussian noises while preserving the defect features. In addition, this paper introduces an optimized CLAHE algorithm to increase the contrast and clarity of noisy images. Finally, a high-precision crack detection algorithm based on deep learning is developed using the latest techniques to enhance accuracy in detecting cracks under different and challenging conditions. A significant highlight of this research is the creation of a real-world augmented image environment specifically designed for deep learning algorithms, tailored to address challenges in pipe inspection and defect detection. This study addresses the significant research gaps in achieving high accuracy, reliability scores, and repeatability in detecting cracks in pipeline inspections Unlike existing methods that require extensive training time and large datasets, our model achieves high performance in just 50 epoch, making it specially designed for real-time business applications.

Comparison of Existing Works:

To provide clarity on the contributions of this paper, Table 1 compares the proposed methods with existing techniques in terms of their advantages and limitations.

The proposed framework includes optimized denoising, contrast enhancement, and deep learning-based detection, addressing the research gaps identified above, making it an effective approach toward improving the precision, reliability, performance and efficiency of crack detection systems in NDT. This paper presents a comprehensive evaluation of the methods proposed and their impact on bridging the identified research gaps.

Table 1. Comparison of Existing Works

Technique	Advantages	Limitations
Traditional Denoising	Simple and computationally efficient	Struggles with preserving fine defect details under heavy noise conditions
CLAHE	Enhances localized contrast effectively	Limited performance in extreme noise and complex surface textures
Deep Learning (YOLO)	High-speed real-time detection	Generalization issues with diverse datasets and environmental conditions Current methods require extensive training time and resources, as well as large datasets.
Proposed Method	Robust optimized denoising Enhanced contrast critical for identifying cracks in bad environmental conditions. Deep learning YOLO with optimized hyper tuning parameters Real world 2825 augmented images with Novel image processing algorithms. Our model achieves high performance in just 50 epoch, making it specially designed for real-time business applications. Inference model with preprocessing performance is 40-50 fps with T4 GPU, which can be further enhanced by using latest next versions of GPU.	Consistent Confidence Score and precision under varied conditions while preserving details. Unusual crack patterns or extremely rare defects not represented in the training dataset.

2 Methodology

It thus requires an automated solution to crack detection in pipes as the limitations of manual inspection. Automation can enhance the accuracy, speed, and reliability of the process by advanced imaging techniques and algorithms of machine learning. It can minimize the human error element, facilitate real-time monitoring, and ensure that minute defects are detected well before they escalate into major failures.

It will involve the usage of highly advanced noise reduction, and robust image enhancement algorithms for the proper detection of advanced methods for the precise methods of crack detection. Then, it needs to undergo software optimization in order for it to work in a real-time environment with much accuracy for timely identification before a critical failure occurs in industries with their pipelines. Ensuring the deployment of reliable crack detection systems is the overall key in NDT. Since traditional inspection methods are slow and labor-intensive, automated detection techniques based on deep learning are quite necessary. For the transfer learning technique to crack detection of concrete structures, various YOLO models are put to the test to improve detection performance. It highlights the significant point that a little cracking detection in engineering structures such as bridges, buildings, and roads may be considered hazards to public safety since these infrastructures received some thinning in terms of structure. The study concludes that YOLOv10 is the model best suited for highly speedy and accurate crack detection of concrete structures. Such findings are a boon for automated deployment of structural health monitoring systems for reducing manual inspection load and improving safety and efficiency in infrastructure maintenance[1].

The major steps involved in crack detection from X-ray pipe images are: Noise Detection, Noise Filtering, Contrast Enhancement, and Defect Detection. These steps guarantee high accuracy in crack detection and superior image quality for the optimization of relevant machine learning models such as YOLOv10. The following is a detailed process flow for each of these steps.

- Noise Detection: Starting from the identification of the noise types affecting the X-ray image, they may include salt-and-pepper noise and Gaussian noise. Noise detection is crucial to ensure that the next filtering step improves noise without damaging any important structural details.
- Noise Filtering: If noise is detected, then after detection of the noise type, the corresponding filtering processes are applied:
- Median filter: Useful for salt-and-pepper noise. This is because it effectively removes isolated noise pixels while conserving edge details.
- Bilateral filter: The most commonly used filter for Gaussian noise as it conserves edge information while effectively reducing noise. This filter applies a weighted average to each pixel's neighbors, with weights determined by spatial distance and intensity difference.
- The parameters of the filters are therefore optimized in real-time based on entropy (for clarity) and edge preservation (using Canny edge detection). In this way, the filter parameters are iteratively adjusted depending on the findings from these measures of image quality.
- Contrast Enhancement: Post Noise Removal, Contrast Limited Adaptive Histogram Equalization (CLAHE) is applied, which amplifies image contrast. CLAHE makes defects (like cracks) visible, whereas it makes the critical features important for defect detection visible.
- Defect Detection: Last, a YOLOv10 (You Only Look Once) model performs defect detection. YOLOv10 is a state-of-the-art real-time object detection model applied to the enhanced image to detect cracks or defects in the pipe.

2.1 Image Acquisition Setup

A general NDT setup to inspect steel pipes for cracks requires a source of radiation, say a gamma or X-ray source. These are positioned around a pipe and are used in checking integrity. Figure 1 provides complete Image Acquisition Setup details. Pulse Xray with different kV is considered and data is augmented as per reference captured images.

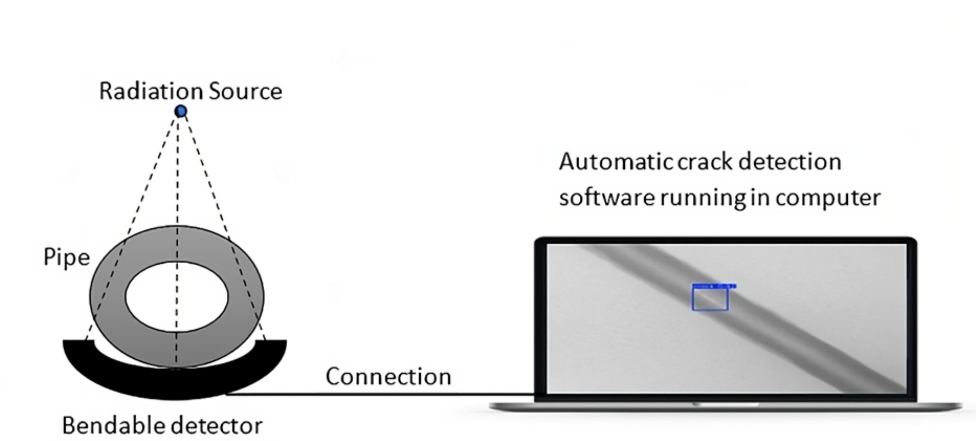

Fig 1. Complete Image Acquisition Setup

The pipe is usually encased in a collimated detector that captures the radiation that passes through or is scattered by the pipe material. The detected radiation data are processed by a computer using specialized software, such as Radiographic Testing (RT) analysis tools, to produce images or scans of the pipe's internal structure. This software can scrutinize the images obtained for cracks and voids, using algorithms to improve clarity and detect defects in terms of size, shape, and position relative to the surface of the pipe, making it clear where structural weaknesses may be. Safety precautions, in terms of shielding and procedures, are necessary for operators as well as the environment when working with sources of radiation. A multi-level novel deep learning based optimized algorithm is developed which includes an optimized denoising algorithm, optimized CLAHE algorithm and optimized YOLOv10 deep learning crack detection algorithm. Figure 2 provides high level details of the proposed multi-level optimized crack detection algorithm.

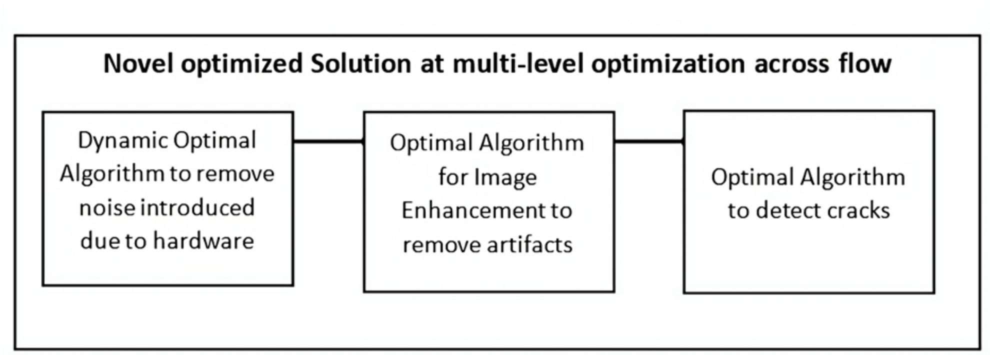

Fig 2. Proposed multi-level optimized crack detection algortihm

Software plays a crucial role in analyzing data collected from these instruments, providing automated detection and assessment of cracks. The software systems are designed to process complex datasets, generating visual and quantitative insights that help technicians make informed decisions about pipeline maintenance. Through the integration of these components, NDT pipe crack detection provides a reliable solution for safeguarding critical infrastructure while minimizing operational downtime and associated costs.

In the world of non-destructive testing, pipe crack detection, software integration has been an important feature for the overall setup to be effective and accurate. The data acquired from the pipe detector and the radiation sources are all compiled within the software to give an integral analysis of the integrity of the material. It is possible for the software to identify anomalies which might be cracks or other types of defects, due to processing huge amounts of data in real time. Noise removal techniques based on MAP estimation and deep learning have been studied extensively, yet some combined solutions for low light and noise remain pending. The Denoising and Histogram Equalization algorithm restores the low-light images with noise from the bridge cracks for better detection of these cracks and increased accuracy in semantic segmentation. Preprocessing like CLAHE has consistently outperformed Contrast Stretching (CS), Histogram Equalization (HE), and the original images across different YOLO models in terms of performance enhancement during training and testing. The enhancement of images using CLAHE improved the performance of YOLO models significantly[2,3].

Advanced algorithms and machine learning techniques have been used within the software, which enhances the detection capabilities by recognizing patterns that are characteristic cracks and distinguishing them from benign imperfections or noise in data. In addition to crack detection, the software allows for the categorization of crack severity, which is key information in making decisions about what to do with the defects. Traditional concrete crack detection is advancing into advanced AI-driven techniques, focusing on the efficiency, accuracy, and real-time processing capabilities of YOLO models. Deep learning-based methodologies, particularly the YOLO series, have been the focal point for real-time crack detection in concrete structures. Deep learning-based YOLO models find extensive application in real-time concrete crack detection, breaking dependence on manual inspection. The YOLOv10-DECA model balances speed and accuracy well and is a feasible solution for real-time crack detection. The attention mechanism helps improve the focus of the model on important areas and, therefore, decreases false detections. The DECA module is added, which adjusts the weights of the convolution kernel dynamically to optimize feature extraction[4–6].

2.2 YOLOv10 framework for Crack detection

YOLOv10 or "You Only Look Once" version 10 is the new generation of the YOLO series object detection algorithms famous for speed and accuracy. Its inherent architecture allows for simultaneous real-time object detection and classification, making it appropriate for use in applications with immediate analysis needs, including NDT for crack detection. YOLOv10 brings with it several crucial improvements over its predecessors and makes it highly suitable for detecting fine details such as cracks in materials[7,8].

One of the major advantages of YOLOv10 is its capability for advanced feature extraction, which is very important for identifying small cracks that might be left out by less sophisticated models. State-of-the-art convolutional layers and optimized anchor boxes ensure that even tiny anomalies can be detected with high precision. Hypertuning of parameters in YOLOv10 provides for customized adjustments tailored to crack detection, which helps increase sensitivity and specificity.

The real-time processing capability of YOLOv10 significantly enhances operational efficiency. For instance, within industrial setups where non-destructive testing is conducted frequently, one can save time and effort regarding the timely detection of cracks to avoid any delay in regular operation. The immediacy further reduces possible hazards before escalating into critical issues, protecting both structural integrity and human life. A practical real-time solution for concrete crack detection by implementing mixed reality, AI, and edge detection techniques. The solution increases efficiency and accuracy over the traditional methods. The solution combines You Only Look Once (YOLO) for crack detection with the Canny edge detection algorithm for skeleton extraction. The automation via Deep Learning (DL) and Image Processing (IP) is more accurate and efficient but lacks real-time interaction with inspectors [9,10].

2.3 Dataset Preprocessing

The preprocessing of raw datasets concerning x-ray images to detect cracks in them is important before the creation of an efficiency and accuracy model such as YOLOv10. There is high variability and detail in most X-ray images that can benefit machine learning in training, or it might become a disaster without proper handling. This common preprocessing technique applied to mitigate computational load and enhance processing speed reduces the resolution of these images [11–13].

This step helps to balance the quality of the image and the efficiency of the training process, so that the model is able to focus on distinguishing critical features without being encumbered by unnecessary detail. Augmentation is another important aspect of preprocessing that significantly increases the robustness and generalizability of the model. Techniques like rotation, scaling, flipping, and changing brightness mimic real-world conditions and hence let the model learn from a variety of scenarios and might be more robust to changes in input data [14,15].

Augmentation not only enriches the training dataset but also protects the model from overfitting so that it remains fit to adapt to unseen data. These preprocessing steps, implemented with great care, act as a basis to formulate an advanced crack detection system, where YOLOv10 can perform well both in terms of precision and speed in real applications to further enhance the capability and reliability of structural safety evaluations [16,17].

For X-ray images of cracks, a variety of augmentation techniques can be used to achieve this. One of the best ways is to apply geometric transformations such as rotation, translation, and scaling. This helps the model identify cracks from different angles and positions. Another way is to apply random flips to augment the dataset by simulating changes in orientation without altering the intrinsic characteristics of the images. Different imaging conditions and machine settings could also be mimicked by adding variations in the intensity and contrast of images. Deep neural networks (DNNs) for automated detection and localization of cracks are evaluated based on wave propagation patterns. Uses DenseNet (Densely Connected Convolutional Network) to enhance feature extraction [18,19].

Noise addition, Gaussian or salt-and-pepper, can further enhance the model robustness by simulating imperfections that are inherently found in real images. The diverse augmentation techniques together will ensure the development of an accurate and reliable model to be applied to a wide variety of X-ray images for crack detection using YOLOv10 end [20].

The dataset used in this study is publicly available on GitHub for transparency and reproducibility [21].

2.4 Dataset Enhancement

Optimizing CLAHE to Enhance Performance: With optimization techniques, several are adopted for the improvement of Contrast Limited Adaptive Histogram Equalization algorithms so that they may effectively operate for the images on which NDT images would have been taken before defect detection using YOLOv10. It seeks an optimum balance between higher image contrast and lower computation overheads without affecting defect characterization integrity. One possible solution to this is the modification of the clip limit parameter in CLAHE [22–24].

By dynamically tuning the clip limit according to local histogram characteristics of the NDT image, poorly contrasted regions can be enhanced quite effectively without over-amplification of noise. This may also be achieved by optimizing the tile size for CLAHE. There are some considerations in finding a suitable tile size for enhancing local contrast. Indeed, a smaller tile tends to offer better local enhancement, but it may introduce various types of artifacts. Larger tile sizes might fail to enhance smaller features quite effectively [25,26].

A hybrid approach that adjusts tile size to the content in an image region can improve balance in enhancement. The following optimization techniques include parallel processing and GPU acceleration to speed up processing time immensely. Given the parallelism in CLAHE, operations such as histogram calculations and pixel mapping can be carried out in multiple threads or processing units, increasing the speed of execution time [27].

These optimizations together target producing high contrast NDT images, making defect detection by YOLOv10 even more accurate and reliable [28]. The implementation of an optimized CLAHE algorithm in collaboration with YOLOv10 for NDT images is an advanced approach towards enhancing image quality along with detection accuracy. CLAHE is a powerful image processing technique that enhances the contrast of images, making finer details more discernible. In the case of NDT images, subtle defects can be critical, so enhancing contrast is imperative [29].

In the advanced CLAHE algorithm, optimization techniques are put in place to choose, at runtime, the value of the clip limit, as well as the dimensions of the tile grid appropriate for the image. So, the optimization would eliminate enhancement that is too brutal, which might add significant noise, or too modest, which would fail to bring out defects. The picture with enhanced contrast and clear details is more appropriate in this case for the applications of deep learning models like YOLOv10 [30,31].

YOLOv10 can benefit significantly from the increased input images, which contain improved contrast. The CNN within the YOLOv10 framework is able to discern defects from defect-free areas better with increased contrast. With this, the model can detect smaller and more subtle defects that would not be detected in the lower contrast images. This means that the integration of optimized CLAHE with YOLOv10 develops a cohesive pipeline that bolsters the efficacy of the detection of defects, allowing it to provide a sturdy solution to industries relying on NDT for quality control and safety.

3 Results and Discussion

3.1 Results And Impact of The Optimized Approach on Defect Detection Accuracy

The optimal Median filter kernel size is determined through real-time evaluation of entropy and edge preservation:

- Kernel Size: The kernel size is optimized within the range of 3x3 to 9x9, where larger kernel sizes tend to remove more noise but may blur edges.
- Default Optimal Parameters: Best default Kernel Size: 5x5 (chosen after testing multiple sizes and selecting the one that maximizes entropy and edge preservation).

The optimal bilateral filter parameters are selected by iterating through different values for the filter's diameter, sigma color, and sigma space:

- Diameter (d): 3, 5, 7, or 9
- Sigma Color (σc): 25, 50, 75, 100
- Sigma Space (σs): 25, 50, 75, 100

Default Optimal Parameters:

- Diameter (d): 9
- Sigma Color (σc): 75
- Sigma Space (σs): 75

The optimized CLAHE algorithm implemented on NDT images enhances the contrast significantly. Figure 3 provides complete details of the proposed Optimal Image Enhancement Algorithm with the Input and output of the pipe under study. In Figure 3, the input image of a steel pipe on the left shows its respective histogram, while on the right exhibits the CLAHE-applied output image with its respective histogram. Table 2 summarizes the parameters used in CLAHE operation on 16-bit pipe images by changing the block size and slope value. Six combinations of block sizes ranging from 2 to 16 and slope values from 2.0 to 4.5 are used for the enhancement of contrast of images. The experimental results summarized in Table 3 correspond to PSNR and SSIM values for each combination of parameters. Among all configurations, the 16×16 block size with a slope value of 4.5 yielded the highest PSNR (55.07) and SSIM (0.99975), thus indicating better image quality with minimal distortion and high structural similarity. These results indicate that the block sizes and the respective slope values are efficient in optimizing the performance of CLAHE for this application.

Contrast enhancement is critical in the context of defect detection, thus paving the way for a sharp increase in the accuracy of identifying flaws by advanced machine learning models such as YOLOv10. When the improved CLAHE algorithm has been incorporated into NDT images, there is a distinct differentiation of defect areas to the background. Thus, subliminal anomalies are noticed more [32,33].

Table 2. Parameters employed for the CLAHE operation on 16-bit steel pipe images

Block Sizes	Slop Values
2	2.0
4	2.5
6	3.0
8	3.5
12	4.0
16	4.5

Table 3. Corresponding PSNR and SSIM values for each parameter combination for default optimal parameters

Block Size	Slope	PSNR	SSIM
2x2	2.5	36.16634404	0.998698026
2x2	3	36.44302141	0.99877726
2x2	3.5	36.81957711	0.99887765
2x2	4	37.08265504	0.99894491
2x2	2	37.70817351	0.998973767
2x2	4.5	37.45625955	0.999032726
4x4	2	49.17472584	0.999925903
4x4	2.5	50.29968567	0.999938098
4x4	3	45.556343	0.999814687
4x4	3.5	40.18726277	0.999359715
4x4	4	35.45787149	0.998077851
4x4	4.5	32.05448406	0.995739819
6x6	2	40.6212574	0.999406318
6x6	2.5	27.89962081	0.988865586
6x6	3	28.7243139	0.990687078
6x6	3.5	30.3742273	0.993566919
6x6	4	28.22507613	0.989306395
6x6	4.5	29.64661333	0.992198978
8x8	2	37.34987865	0.998674309
8x8	2.5	33.84127508	0.997025111
8x8	3	35.39281391	0.997907972
8x8	3.5	33.34983552	0.996619785
8x8	4	31.48957236	0.994741579
8x8	4.5	32.15632103	0.995438702
12x12	2	34.82705603	0.997515666
12x12	2.5	35.03239212	0.997627089
12x12	3	35.5147108	0.997868491
12x12	3.5	35.65640641	0.997932567
12x12	4	37.40849677	0.998614032
12x12	4.5	39.49635322	0.999138445
16x16	2	37.9115463	0.998746743
16x16	2.5	42.6270342	0.999578859
16x16	3	46.68382047	0.999834363
16x16	3.5	50.03708118	0.999923336
16x16	4	52.6845244	0.999958293
16x16	4.5	55.07846817	0.999975941

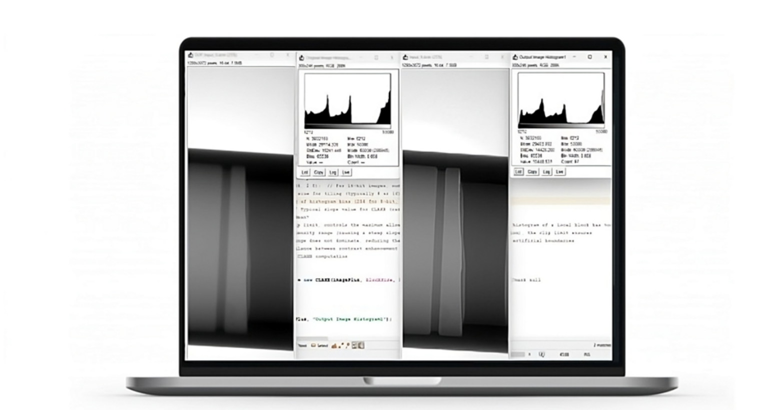

Fig 3. Proposed Optimal Image Enhancement Algorithm with Input and Output details

The quantitative analysis shows that the pre-processing of images has greatly increased the accuracy of YOLOv10 in detecting defects as compared to images without any processing. More contrast in the image can highlight the features associated with a defect, thus enabling easier learning for the YOLOv10 model as it learns from more defined and detailed image data. High precision, recall, and F1 scores are obtained for these defect detection tasks [34,35].

The ability of YOLOv10 is enhanced to detect a wider variety of defect sizes and types, as well as its resistance to variations in lighting and noise in images. In actual practice, this means a reduction in false positives and negatives, which leads to more reliable and timely defects detection. This not only optimizes the inspection process but also significantly reduces the risk of overlooking critical defects that could compromise the structural integrity or functionality of the tested materials. Consequently, the optimized CLAHE algorithm's contribution to contrast enhancement directly translates to enhanced safety, reduced inspection times, and lower operational costs in industrial applications employing NDT imaging for quality control and defect detection [36].

3.2 Process of experimentation

This paper uses the state-of-the-art deep learning algorithm YOLOv10 to train a model for crack detection on pipes. The images were first manually annotated to identify crack regions, which formed the basis of the dataset. Data preprocessing was then applied to improve image quality and model robustness, including denoising and Contrast Limited Adaptive Histogram Equalization (CLAHE) to enhance contrast and reveal subtle crack details.

3.2.1 Training Process for YOLO

The YOLO model was trained using 2,938 crack images of pipes as data, dividing the images into 80% for training, 10% for validation and 10% for testing. The objective of machine learning was to train the model using the sets while the goal of data analysis was to examine the model test output during parameterization. Some of the learning parameters with the set ranges that were optimized included: learning rate, batch size and method of augmentation. Training through experimentation showed that performing 50 epochs produced accurate models without compromising generalization, while 40 epochs were underfitting and 80 epochs were overfitting. A model that could identify fine details was crucial so image size was modified, data augmentation methods which included horizontal flips and contrast scaling were employed, and the optimizer used was AdamW starting with a learning rate of 0. 001.The experiment aimed at training YOLOv10 over several epochs whereby the model learned optimal parameters for weights and bias. The model performance was evaluated on Precision, Recall, F1 score, mean Average Precision (mAP), and single-image detection time.

The evaluation metrics are described as follows:

Precision is calculated in Equation (1) wherein TP represents true positive detection, which means the presence of cracks and FP represents the false positives, which incorrectly identify cracks. Precision expresses the proportion of correctly cracked images out of all cracked images identified.

The following formulas describe the evaluation metrics used:

Precision is calculated as shown in Equation (1), where TP represents true positive detections (correctly identified cracks) and FP represents false positives (incorrectly identified cracks). Precision indicates the proportion of correctly identified cracks out of all identified crack images.

$$Precision\ = TP/(TP\ +\ FP) \tag{1}$$

Recall (Equation (2)) represents the ratio of correctly identified cracks (TP) to all actual cracks in the dataset, with FN as false negatives (cracks missed by the model).

$$Recall = TP/(TP + FN) \tag{2}$$

F1 Score (Equation (3)) provides a harmonic mean of Precision and Recall, useful when high values for both metrics are desired.

$$F1 = 2 \times (Precision \times Recall)/(Precision + Recall) \tag{3}$$

Average Precision (AP) (Equation (4)) captures the global performance by averaging precision across all confidence thresholds, resulting in the mAP (mean Average Precision) offering a comprehensive measure of model accuracy.

$$AP = \Sigma Precision \times \Delta Recall \tag{4}$$

This experimental process establishes a solid framework for real-time crack detection on steel structures, ensuring both high accuracy and efficiency, supported by preprocessing enhancements such as denoising and CLAHE.

3.3 Assessment of the results of experiments

Results seen from training YOLOv10 on crack detection through different epochs, Table 4 provides insights on the model's performance trend. Assessments are based on the differences in precision, confidence score, and recall over the epoch training from 40 to 80 epochs.

Table 4. Results of Deep Learning Algorithm Experiments with different Epochs

Epochs	Precision	Confidence Score	Recall
40	99.4	70 to 77	95.7
50	95.1	88 to 92	100
80	95	80 to 85	95

3.3.1 Epochs 40 - 50: Initial Precision Drop and Confidence Score Increase

- Observation: Between 40 and 50 epochs, we see a drop in precision from 99.4% to 95.1% accompanied by an increase in the range of the confidence score from 70-77 to 88-92 and an improvement in recall to 100%.
- Explanation: This stage shows that the more the model is trained (from 40 to 50 epochs), the more it is confident with its detections, which is why the confidence score range increased. The model learns better generalization of the whole dataset, and hence the overfitting decreases. It is then able to detect a much wider range of cracks, which in turn increases recall.
- Precision Drop: The small drop in precision may suggest that the model is making more confident predictions overall but also some incorrect predictions (false positives), thus reducing precision slightly. This trade-off often happens as the model tries to balance recall (correctly identifying all true positives) and precision.
- Increased Confidence Scores: Increased confidence scores imply that the model has increased its confidence in the predictions, meaning it has learned the characteristics of cracks better.
- Improved Recall: the boosting to 100% recall means that it is now capable of recalling all actual cracks, including not missing any, probably because of better learned representation from increasing epochs.

3.3.2 Epochs 50 - 80: Plateaued Confidence Scores and Reduced Recall

- Observation: When training progresses to 80 epochs, the range of the confidence score stabilizes between 80-85 while having a minor drop in recall (from 100% to 95%) and a slight decrease in precision (from 95.1% to 95%).
- Explanation: As training progresses, the model begins to overfit, where it becomes too sensitive to specific features in the training data and loses its ability to generalize. This phase usually reflects stabilization or a slight decline in performance metrics:

- Confidence Score Plateau: The confidence score stabilizing around 80-85 indicates that the model is not gaining additional certainty with continued training. This means that the model has reached a point where it no longer improves in discerning crack features and is not further refining its confidence levels.
- Decrease in Recall: This slight decrease in recall means that the model is perhaps overfitting, resulting in missing some true positives that it had previously picked up. The model narrows down with increasing training time and may not be able to generalize to all cracks in the validation set, thus missing a few detections.
- Low Precision Decrease: The slight loss in precision could be because the model has become too confident with its predictions and sometimes generates false positives, which marginally affect precision. This may be due to the model ”locking in” some of the learned features that are unique to the training data, resulting in less flexibility in handling unseen or slightly different crack characteristics.

3.3.3 Balancing Precision, Recall, and Confidence Score

Based on these observations, an ideal training period appears to be roughly 50 epochs, wherein the model maintains high recall, reasonable precision, and strong confidence scores without signs of overfitting. Further training does not confer significant benefits in confidence but does begin to negatively impact recall and slightly reduce precision.

Practical Implication: Training YOLOv10 up to 50 epochs balances performance well between identifying all relevant cracks with high recall, reasonable precision, and confident detections. Further training beyond this point provides diminishing returns and may introduce minor degradation in generalization performance. Figure 4 provides details of real time Optimal Confidence Score of the deep learning crack detection algorithm.

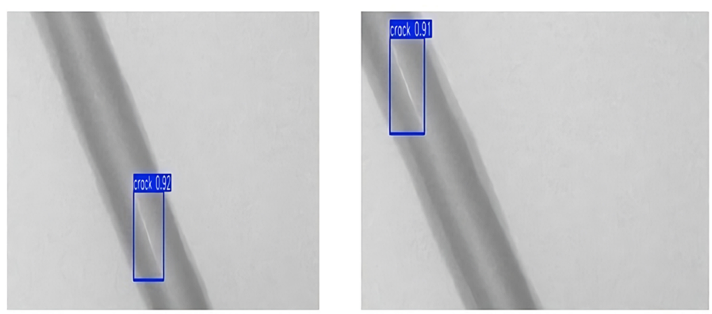

Fig 4. Real time Optimal Confidence Score

3.3.4 Normalized Confidence Score

- The normalized confusion matrix is a visual summary of the performance of the classifier in terms of the proportion of correct or incorrect predictions for each class. For crack detection, the entries are interpreted as follows:
- Layout and Values of the Matrix: Figure 5 provides complete details of the Normalized Confusion Matrix. Below is a detailed explanation:
- Top left (0.96) This represents the percentage of instances that were correctly classified as cracks, that is True Positives.
- Top right (1.00): It represents the percentage of crack instances wrongly classified as background or False Negatives. Being a normalized matrix, ”1.00” here may imply rounding from a tiny proportion but it suggests that none of the true background examples got misclassified as cracks
- Bottom left (0.04): It denotes the percentage of background instances that got mistakenly classified as cracks or False Positives.
- Bottom right (no value): True negatives, that is, the percentage of background instances correctly classified as background. Since there is nothing here, it could mean no true background instances were present or misclassified and could possibly simplify the matrix into a binary form.
- High True Positive Rate (0.96): The model correctly identifies cracks 96% of the time, thus showing strong accuracy in the detection of cracks.
- Low False Positive Rate (0.04): That means only 4% of non-crack instances are classified as crack, which shows the model’s accuracy in distinguishing between crack vs. background features.

- False Negative and True Negative Indicators: As there is no misclassification of crack as background or vice versa, the matrix implies that the model is faced with very few ”background” elements or distinguishes them very well.

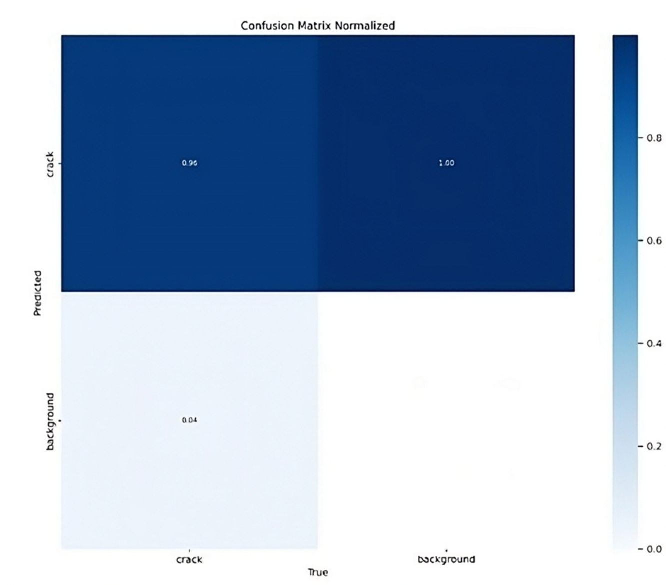

Fig 5. Normalized Confusion Matrix

3.3.5 Implications of Crack Detection

Strong Detection: It is appropriate for environments where there is a high need for reliable crack detection with a minimum of false positives.

Balanced Performance: The model is seen to have learned enough detail to focus reliably on cracks, with little noise from background errors, in the sense that it misclassifies at a low rate into the background class.

This configuration reflects a reliable model with a strong focus on accurate crack detection, with potential utility in real-time or high-stakes inspection environments where both sensitivity and specificity are required. Figure 6 and Figure 7 provide complete details of Train and Validation losses for Epochs 40 and 50.

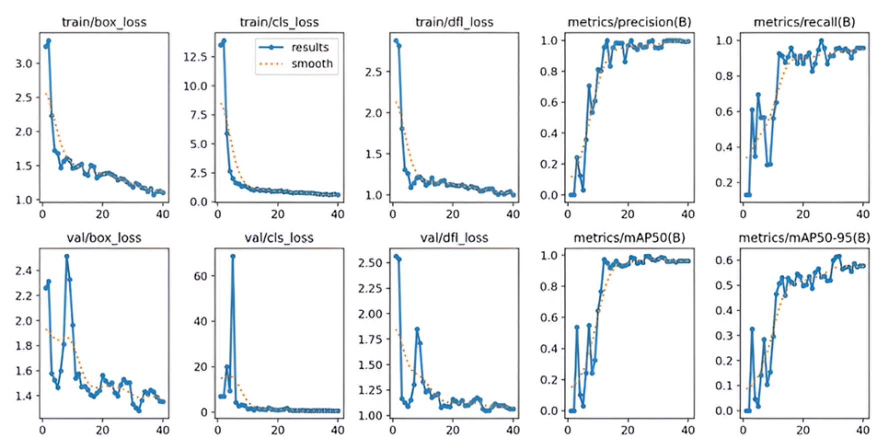

Fig 6. Train and Validation losses Epochs 40

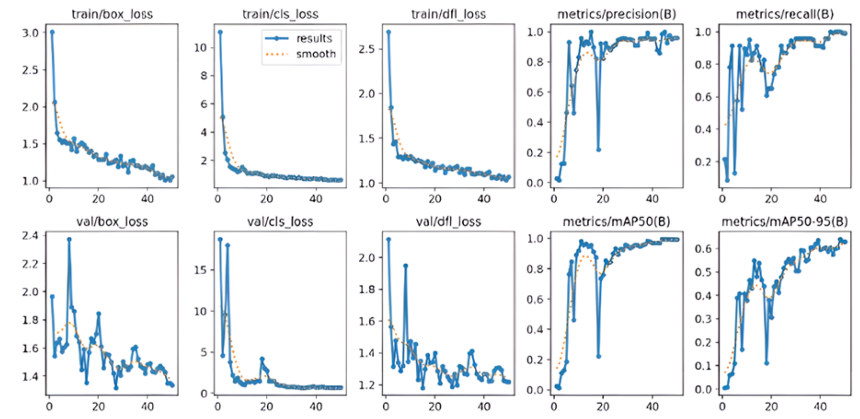

Fig 7. Train and Validation losses Epochs 50

- **Box Loss (CIoU Loss):** Measures the difference between the ground truth and the expected bounding boxes by Box Loss. It optimizes the size, position, and shape of the bounding box.
- **Class Loss (Binary Cross-Entropy):** Shows the ability of the model to classify the appropriate categories for the objects.
- **Distribution Focal Loss:** Normally, conventional object detection algorithms usually predict the box's x, y, width, and height as continuous values. However, DFL promotes localization by predicting a distribution of the coordinates of the bounding box across a range of discrete places instead of only regressing those continuous values. Using this technique, the model could focus more effectively on fine-grained information for the localization of objects.

In general terms, the model is generalizing well to the validation set if the validation loss tends to zero faster than the related training losses. Since validation loss is a measure of how well the model will perform on unseen data, the fact that it is getting near zero early on is encouraging: it suggests that the model is picking up significant and widely applicable features from the training data.

The term "distribution focal loss" or dfl_loss refers to a type of focal loss that enhances model performance in the case of unbalanced training data. In particular, the class imbalance that occurs while training on datasets containing extremely rare items is addressed via distribution focused loss. The network will find it challenging to learn to correctly classify items of a particular class when there are few examples of that class in the training set. The distribution focal loss helps to ensure that the model correctly identifies these rare objects. By maximizing the three important components of object detection: localization (bounding box), classification (object class), and abjectness (confidence), these loss functions combine to improve performance.

3.3.6 Real time pipe Crack Detection performance

As per the current research, YOLO deep learning model for cracking detection in steel pipe had both GPU & CPU utilization to check for real-time performance. Using a Tesla T4 GPU (CUDA: 12.2), 15,102 MiB of memory was applied while achieving an approximate inference time of 16.6 ms per image at resolutions of 448 x 640 pixels. This GPU-based model achieved processing speeds of around 60 fps which is affordable to real-time applications. The A100 offers a 20 times performance improvement compared to its predecessors, making it a significant improvement for real-time inference and high-performance computing applications. Upgrades to a more advanced GPU such as the A100 or even the latest RTX 50-series can bring significant improvements in processing speed and are better suited for demanding real-time applications.

Performance Impact Estimation:

Given the current inference speed of 60 FPS (via the Tesla T4 GPU, with an image processing time of 16.6 ms), the estimated range of processing time for performing its preprocessing is seen as follows:

- Median Filter (3x3 kernel): ~1-2 ms
- Bilateral Filter (5x5 kernel, optimized parameters): ~2-4 ms
- Entropy and Edge-based Parameter Selection: ~1-3 ms (if computed efficiently)
- CLAHE Application: ~2-3 ms

Adding up these operations would further fling in 5-10 ms processing time per image, leading to a drop in the FPS in the range of 40-50 FPS, considering the parameters and efficiency of the hardware.

Optimization Strategy for Retaining Real-Time Performance:

In order to prevent excessive computational delays, a hybrid parameter selection strategy can be implemented:

Default Offline Parameters: Precomputed parameters (based on analysis of the dataset) would be set up in the beginning. These parameters have been optimized for average noise conditions.

Adaptive Real-Time Tuning (Conditional Execution): If there have been drastic changes in entropy or edge clarity, new filter parameters will be computed on-the-fly. Otherwise, the best parameters already stored would be reused to minimize the computational overhead.

Adaptable contrast enhancement (CLAHE) via local contrast variance and edge density is considered here so as not to over-process while obtaining real-time efficiency. With the computed local variance across different regions of the image, if the variance is finished, boosting the clip limit will enhance weak contrast. In contrast, if the variance appears high, a decrease in the clip limit would assist in avoiding over-enhancement. Edge density is obtained using Canny edge detector. When the edge density is high, it means the block size needs to be increased to avoid the introduction of unnecessary artifacts. When the edge density is low, this will mean using a smaller block size to enhance fine details.

Efficient Implementations:

The GPU could be used to accelerate filtering (OpenCV CUDA, TensorRT, PyTorch) to further minimize latency. While the previous frame is working through inference, an asynchronous CLAHE and filtering operation would take place. Regarding filter down sampling, a downsize value probably of 50% could be used for evaluating entropy and edge detection calculation.

By this optimized strategy, the pipeline for real-time crack detection would enhance the visibility of defected areas while reducing noise artifacts for their representation, having high FPSs (~50 FPS or more) in that way. The adaptive noise filtering will ensure our consideration.

This output can be deployed in real-time automated systems integrated into GPU based programs at inspection machines, robotic systems, or drones for structural pipe health change detection. It can be reliably used for crack detection due to the high confidence levels and automated result saving, thus integrating well into automated maintenance systems and high-performance defect detection in industrial systems.

3.3.7 YOLO-CNN Optimization

In the YOLO framework, the CNN (Convolutional Neural Network) is the backbone of the model, responsible for feature extraction from input images. Several hyperparameters directly impact the Convolutional Neural Network (CNN). Optimization in YOLO training implementations attempts to enhance the efficiency of techniques used in the modeling process. This training employs Automatic Mixed Precision (AMP), which reduces the computational load by dynamically managing numerical precision. The optimizations should minimize memory use and also speed up inference with no effect on accuracy. The AdamW optimizer is selected automatically, which balances the strategies for adjusting the learning rate along with weight decay to avoid any case of overfitting. With 225 layers (11,135,987 parameters/28.6 GFLOPs), the model architecture strikes a balance between computational efficiency and high performance in detection. Some of the hyperparameters that influence the convergence speed and generalization for more effective training processes are the momentum (0.937), weight decay (0.0005)- which allow for learning rate scheduling hyperparameter experiments as well. Training used lr0=0.01, and the optimizer automatically set the best learning rate as AdamW(lr=0.002). The learning rate is crucial in CNN optimization as it controls how much the model updates weights per iteration. A high learning rate may lead to divergence, while a low one may slow convergence.

The convolutions and up sampling layers are structured inside the model to minimize redundant computation so that feature extraction is optimized. The final results of training indicate very high precision (0.984), recall (0.957), and mean average precision (0.989 at IoU 0.5), reflecting the successes entrenched in these optimizations. Finally, the lower loss values recorded (box loss: 1.06, class loss: 0.5982, DFL loss: 0.9837) by the final epoch highly attested to the power of mathematical optimization techniques of CNN in this defect detection work.

3.4 Comparison with Other Studies

Our study targeted realistic data augmentation techniques for enhancing the performance of steel pipe crack detection models. For better contextualization of our results, we compared them with a leading previous study, "Deep Learning Based Steel Pipe Weld Defect Detection" by Author et al.

Below is a detailed comparison:

Data Augmentation Approaches:

The previous study augmented their 119 crack samples to generate 1071 augmented samples using general augmentation techniques such as light change, random rotation, cutout, Gaussian noise addition, horizontal flipping, and random adjustments to saturation, contrast, and sharpness. Our study, however, began with a smaller dataset of 113 crack samples but generated 2825 real world augmented samples using more realistic and domain-specific augmentation techniques. Significant focus and effort are given to developing real world images.

These included overlaying artificial cracks, adding simulated rust or corrosion patterns, applying affine and perspective transformations, making elastic deformations, enhancement of edges, speckle noise, and blur to simulate some real-word scenarios. This generated wide and realistic samples of what indeed happens in real-world inspections.

Model Evaluation:

The prior work attained a score of 0.69 for confidence and a mAP of 99.02% at the epoch 218. However, when the model was prolonged in training, it suffered from overfitting because, after 633 epochs, it declined MAP to 98.71%.

The results here with a confidence score of 0.92 are consistently over 99.25% mAP without encountering signs of overfitting in the results with extended training. That explains how the augmentation enhances the generalization ability in a model.

Precision, Recall, and F1 Score:

Even though the earlier work shows a perfect score of 1.0 for precision, recall, and F1, our model had the same scores. Nevertheless, the strength of our approach is that it outperforms in more difficult and practical scenarios because of the introduction of augmented data that mimics the imperfections of real-world settings.

Novelty and Innovations in Data Augmentation:

The novelty of our work involves the adoption of realistic and domain-specific data augmentation techniques, which greatly improve both the diversity and quality of training data. Our work is different from other prior studies that have utilized more general augmentation approaches because of the unique characteristics of crack detection in steel pipes, such as rust, stains, noise, and perspective variations.

Furthermore, the outcomes underscore that our augmentation strategy successfully avoids overfitting, which was a problem in earlier works. It thus brings out the innovation in our methodology and can be beneficial for enhancing crack detection accuracy in practical scenarios.

4 Conclusion

The presented study introduces a highly optimized framework for real-time crack detection on steel surfaces and gives important advancements over methodologies so far. Our research incorporates optimal denoising with CLAHE algorithm coupled with fine-tuned hyperparameters to establish an efficient data preprocessing pipeline, with optimizations that improve clarity and the ability of models to generalize, thus overcoming the tendency to overfitting, making the detection even more robust.

Key Contributions and Novelty:

Optimized Deep Learning Framework: By leveraging the YOLOv10 architecture with hyperparameter optimization, our method achieves a remarkable mAP (mean average precision) exceeding 99.25%, consistently outperforming earlier iterations. Unlike previous studies that required extensive training cycles, our model achieves high precision, confidence Score, and recall with only 50 epochs, making it ideal for real-time industrial applications.

Quantitative Achievements: The proposed model classifies cracks with a confidence score of 0.92 and does not suffer from overfitting even during extended training. It is faster and more efficient compared to traditional methods, and hence a benchmark for future real-time crack detection models.

Data Augmentation with Realistic Simulations: It is a strong data enhancement strategy, including augmented simulations of real-world crack environments that will make the model versatile for various conditions. Such conditions include adding Gaussian noise, perspective transformations, and edge enhancement to simulate real-world conditions of the pipes and overfitting problems are significantly reduced.

Strengths and Weaknesses:

The strengths of this study are in the way it combines advanced preprocessing and augmentation techniques with optimized deep learning for industrial defect detection. Another way our method differs is that it integrates real-world crack simulation during augmentation, ensuring a robust and scalable model. However, a major limitation is the model's sensitivity to highly unusual crack patterns or defects not represented in the training dataset. While our augmentation strategy addresses many real-world scenarios, the model may still struggle with extremely rare defects.

Improvements and Open Questions:

Scope of the study is limited to two types of noise salt-pepper and Gaussian noise, further research can include different types of noise related to NDT environment. Future work would be the integration of few-shot learning techniques, so that quality control personnel could manually tag unique or rare crack patterns, ensuring continuous model improvement. Domain adaptation techniques to handle variations in imaging conditions across different production lines could also be further explored to enhance the versatility of the model. There is still an open question on how best to incorporate multimodal data, such as thermal or ultrasonic imaging, into the framework for more comprehensive defect analysis.

Prospects and Recommendations:

The proposed framework provides a solid foundation for advancing the automation of production. Future researchers are advised to extend this work to other industrial materials beyond pipes to ensure more general applicability. Moreover, explainability techniques can help interpret model decisions, increasing trust and usability in industrial environments.

References

1) Sohaib M, Arif M, Kim JM. Evaluating YOLO Models for Efficient Crack Detection in Concrete Structures Using Transfer Learning. *Buildings*. 2024;14(12). Available from: https://doi.org/10.3390/buildings14123928.
2) Qiu G, Tao D, You D, Wu L. Low-illumination and noisy bridge crack image restoration by deep CNN denoiser and normalized flow module. *Scientific Reports*. 2024;14. Available from: https://www.nature.com/articles/s41598-024-69412-5.
3) Chen RC, Dewi C, Zhuang YC, Chen JK. Contrast limited adaptive histogram equalization for recognizing road marking at night based on YOLO models. *IEEE Access*. 2023;11:92926–92942. Available from: https://ieeexplore.ieee.org/document/10232983.
4) Mustafa WA, Yazid H, Alquran H, Al-Issa Y, Junaini S. Significant effect of image contrast enhancement on weld defect detection. *PLOS ONE*. 2024;19(6). Available from: https://pmc.ncbi.nlm.nih.gov/articles/PMC11213325/.
5) Zhang C, Peng N, Yan J, Wang L, Chen Y, Zhou Z, et al. A Novel YOLOv10-DECA Model for Real-Time Detection of Concrete Cracks. *Buildings*. 2024;14(10). Available from: https://doi.org/10.3390/buildings14103230.
6) Bassir D, Chang H, Majak J. Real-Time Crack Detection on Concrete Using Non-Destructive Approaches Based on YOLO Series. In: and others, editor. Advances in Machinery, Materials Science and Engineering Application X;vol. 58. 2024;p. 820–826. Available from: https://www.researchgate.net/publication/385061768_Real-Time_Crack_Detection_on_Concrete_Using_Non-Destructive_Approaches_Based_on_Yolo_Series_Review.
7) Hena B, Wei Z, Castanedo CI, Maldague X. Deep learning neural network performance on NDT digital X-ray radiography images: Analyzing the impact of image quality parameters-An experimental study. *Sensors*. 2023;23(9). Available from: https://www.mdpi.com/1424-8220/23/9/4324.
8) Sidar I, Davidson T, Kronman A, Mor L, Levy I. Endoscopic image enhancement using contrast-limited adaptive histogram equalization (CLAHE) implemented in a processor. ES2818174T3. Published. 2016. Available from: https://patents.google.com/patent/ES2818174T3/en.
9) Cahyani S, Sari AK, Harjoko A. Underwater quality enhancement based on mixture contrast limited adaptive histogram and multiscale fusion. *International Journal of Advanced Computer Science and Applications*. 2024;15(7):552–561. Available from: https://thesai.org/Publications/ViewPaper?Volume=15&Issue=7&Code=ijacsa&SerialNo=63.
10) Shojaei D, Jafary P, Zhang Z. Mixed Reality-Based Concrete Crack Detection and Skeleton Extraction Using Deep Learning and Image Processing. *Electronics*. 2024;13(22). Available from: https://doi.org/10.3390/electronics13224426.
11) for geeks G. CLAHE histogram equalization - OpenCV. 2023. Available from: https://www.geeksforgeeks.org/clahe-histogram-eqalization-opencv/.
12) Build Vision Models with Roboflow. 2024. Available from: https://docs.roboflow.com/.
13) Khaledyan D, Marini TJ, Baran TM, Connell AO, Parker K. Enhancing breast ultrasound segmentation through fine-tuning and optimization techniques: Sharp attention UNet. *PLOS ONE*. 2023;18(12). Available from: https://doi.org/10.1371/journal.pone.0289195.
14) Haddadi YR, Mansouri B, zohra Driss Khoudja F. A novel medical image enhancement algorithm based on CLAHE and Pelican optimization. *CoLab*. 2023. Available from: https://colab.ws/articles/10.21203/rs.3.rs-2443705/v1.
15) Wang AL, Hansen MH, Lai YC, Dong J, Xie KY. Improving orientation mapping by enhancing the diffraction signal using Auto-CLAHE in precession electron diffraction data. *Microstructures*. 2023. Available from: https://www.oaepublish.com/articles/microstructures.2023.27.
16) Anwar A. A comprehensive performance analysis of transfer learning optimization in visual field detection. 2024;p. 1–26. Available from: https://www.academia.edu/118903100/A_Comprehensive_Performance_Analysis_of_Transfer_Lea.
17) Zetec. C-Scans in Ultrasonic Testing: The Importance of Clear Visualization in Analysis. 2024. Available from: https://www.zetec.com/blog/c-scans-in-ultrasonic-testing-the-importance-of-clear-visualization-in-analysis.
18) Moreh F, Lyu H, Rizvi ZH, Wuttke F. Deep neural networks for crack detection inside structures. *Scientific Reports*. 2024;14. Available from: https://www.nature.com/articles/s41598-024-54494-y.
19) NDT Equipment Selection: A Complete Guide. 2024. Available from: https://www.ndtproducts.ca/a-complete-ndt-equipment-selection-guide.
20) What Is Non-Destructive Testing (NDT) And How Is It Used. 2024. Available from: https://limblecmms.com/blog/non-destructive-testing-ndt.
21) Github. 2024. Available from: https://github.com/lucky999-lab/dataset.
22) Petrosync. What is NDT? A Complete Guide to Techniques & Benefits. 2024. Available from: https://www.petrosync.com/blog/what-is-ndt.
23) Zhou J, Ning J, Xiang Z, Yin P. ICDW-YOLO: An efficient timber construction crack detection algorithm. *Sensors (Basel)*. 2024;24(13). Available from: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC11244569.
24) Piping & Pipeline Monitoring. 2024. Available from: https://www.mistrasgroup.com/how-we-help/monitoring/piping-pipelines.
25) Labelvisor. 2024. Available from: https://www.labelvisor.com/applications-of-yolov10-in-object-detection.
26) YOLO Object Detection Explained: Evolution, Algorithm, and Applications. 2024. Available from: https://encord.com/blog/yolo-object-detection-guide.
27) Non-Destructive Crack Detection Vessel Testing. 2023. Available from: https://making.com/equipment/non-destructive-testing-vessel-crack-detection.

28) Why you need quality data labeling in infrastructure crack detection. 2024. Available from: https://blog.cloudfactory.com/quality-data-labeling-for-infrastructure-crack-detection.
29) Haddad A. Image-Based Crack Detection Methods. 2024. Available from: https://encyclopedia.pub/entry/13326.
30) Improving Data Quality Using End-to-End Data Pre-Processing Techniques in Encord Active. 2024. Available from: https://encord.com/blog/enhancing-data-quality-in-computer-vision/.
31) What is image classification? Basics you need to know. 2024. Available from: https://www.superannotate.com/blog/image-classification-basics.
32) Chakurkar PS, Vora D, Patil S, Mishra S, Kotecha K. Data-driven approach for AI-based crack detection: techniques, challenges, and future scope. *Front Sustain Cities*. 2024;5. Available from: https://www.frontiersin.org/journals/sustainable-cities/articles/10.3389/frsc.2023.1253627/full.
33) Clarke H. Mastering data preprocessing for AI: Elevating model performance. 2024. Available from: https://www.harrisonclarke.com/blog/mastering-data-preprocessing-for-ai-elevating-model-performance.
34) Carpentries. Introduction to Image Data. 2024. Available from: https://carpentries-incubator.github.io/intro-image-classification-cnn/02-image-data.html.
35) AI Going beyond ChatGPT: How Data Augmentation is reshaping Orthopedics. 2024. Available from: https://blog.peekmed.com/data-augmentation-orthopedics.
36) Desternes J. Data augmentation for medical image analysis in deep learning. 2024. Available from: https://www.imaios.com/en/resources/blog/ai-for-medical-imaging-data-augmentation.