

Article

AI-Driven Crack Detection for Remanufacturing Cylinder Heads Using Deep Learning and Engineering-Informed Data Augmentation

Mohammad Mohammadzadeh ^{1,*} , Gül E. Okudan Kremer ² , Sigurdur Olafsson ^{1,*}  and Paul A. Kremer ³

¹ Department of Industrial and Manufacturing Systems Engineering, Iowa State University, Ames, IA 50011, USA

² School of Engineering, University of Dayton, Dayton, OH 45469, USA; gkremer2@udayton.edu

³ Department of Civil, Construction and Environmental Engineering, Iowa State University, Ames, IA 50011, USA; pkremer@iastate.edu

* Correspondence: moha95@iastate.edu (M.M.); olafsson@iastate.edu (S.O.)

Abstract: Detecting cracks in cylinder heads traditionally relies on manual inspection, which is time-consuming and susceptible to human error. As an alternative, automated object detection utilizing computer vision and machine learning models has been explored. However, these methods often face challenges due to a lack of sufficiently annotated training data, limited image diversity, and the inherently small size of cracks. Addressing these constraints, this paper introduces a novel automated crack-detection method that enhances data availability through a synthetic data generation technique. Unlike general data augmentation practices, our method involves copying cracks from one location to another, guided by both random and informed engineering decisions about likely crack formations due to cyclic thermomechanical loads. The innovative aspect of our approach lies in the integration of domain-specific engineering knowledge into the synthetic generation process, which substantially improves detection accuracy. We evaluate our method's effectiveness using two metrics: the F2 score, which emphasizes recall to prioritize detecting all potential cracks, and mean average precision (MAP), a standard measure in object detection. Experimental results demonstrate that, without engineering insights, our method increases the F2 score from 0.40 to 0.65, while maintaining a stable MAP. Incorporating detailed engineering knowledge further enhances the F2 score to 0.70 and improves MAP to 0.57, representing increases of 63% and 43%, respectively. These results confirm that our approach not only mitigates the limitations of traditional data augmentation but also significantly advances the reliability and precision of crack detection in industrial settings.

Keywords: defect detection; deep learning; computer vision; data augmentation; engineering knowledge integration; remanufacturing



Citation: Mohammadzadeh, M.; Kremer, G.E.O.; Olafsson, S.; Kremer, P.A. AI-Driven Crack Detection for Remanufacturing Cylinder Heads Using Deep Learning and Engineering-Informed Data Augmentation. *Automation* **2024**, *5*, 578–596. <https://doi.org/10.3390/automation5040033>

Received: 16 September 2024
Revised: 5 November 2024
Accepted: 15 November 2024
Published: 20 November 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Remanufacturing is essential to implementing circular economy practices because it effectively reduces material consumption, energy consumption, and waste. This is accomplished by retaining the value of extracted and refined raw materials. The remanufacturing process includes several steps, such as cleaning, inspection, and reconditioning, designed to restore a used core product to the specifications of a brand-new product [1]. The inspection step evaluates the condition of the core product and identifies any defects or deviations from design specifications, such as size and location [2]. Quality inspection is crucial to achieving remanufacturing goals, such as customer satisfaction, reducing rework due to inadequate inspection, and making informed decisions regarding the viability of remanufacturing a used product or core.

The inspection procedure can be carried out manually, semi-automatically, or automatically. Manual inspection, which is labor-intensive and time-consuming, is frequently

necessary for remanufacturing to ensure 100 percent inspection, build trust with secondary users, and increase the manufacturer's profitability [3]. However, inspector experience, skills, the timing of inspection (such as a night shift versus morning shift), and inspection location can have a substantial effect on the errors [4–6].

As inspection tools and techniques continue to advance, fully automated systems utilizing machine learning (ML) techniques and machine vision systems are being developed to assist human inspectors in remanufacturing processes. Modern machine vision technology enables the detection of various defects of interest with high resolution and precision without significantly extending the inspection time. The utilization of image processing techniques for inspection and quality control, particularly in defect detection, has become commonplace [7]. Automation plays a crucial role in the modern manufacturing industry, and image processing techniques for inspection and quality control have become widespread. As the demand for even greater accuracy and complexity in quality control tasks grows, deep learning has emerged as a transformative force in this field. Deep learning-based inspection systems incorporating layered, multi-layer neural networks are gaining popularity as they efficiently extract pertinent features to characterize data patterns. Convolutional neural networks (CNNs), a class of deep learning architectures, are inspired by the visual perception of living organisms [8]. Various computer vision tasks, such as image recognition, object detection, and localization, have demonstrated their efficacy [9]. By simulating the complex processes of visual perception, CNNs provide innovative solutions in artificial intelligence, enabling machines to accomplish remarkable feats in comprehending and interpreting visual data. Object detection algorithms based on CNN can be broadly divided into one- and two-stage models.

One-stage models provide real-time responses without needing object detection steps in advance, making them suitable for quick predictions. An example is YOLO v5, a one-stage detector model from the You Only Look Once (YOLO) family of computer vision models. It is renowned for its rapid image processing capabilities, outperforming the models above by applying inline transformations to the base training data, incorporating a broader range of semantic variation, and calculating loss functions to maximize the mean average precision (MAP) objective [10].

On the other hand, two-stage models employ preliminary stages in which significant image regions are identified before object classification. Faster R-CNN ResNet101 is a high-precision, two-stage object detection model that divides the region proposal and classification steps into separate stages to locate and classify objects. Although it achieves greater prediction accuracy than other methods, its prediction speed needs improvement [11].

CNN application in crack detection has shown promise in terms of accurately identifying cracks in various structures. However, the performance of deep neural networks is highly dependent on the availability of an extensive and well-annotated training dataset. Inadequate training data can lead to overfitting and subpar predictive performance [12]. The performance of deep neural networks is significantly enhanced by increasing the quality and quantity of training data. Despite the availability of numerous open-source datasets for crack detection in road surfaces [13] and concrete structures, the lack of datasets designed specifically for cylinder head surface cracks poses a significant barrier to accurately detecting such cracks on the said surfaces. Publicly accessible datasets currently have limitations when applied to detecting cylinder head cracks. Notably, cylinder head cracks are typically much smaller than those on concrete or asphalt surfaces. Detecting these cracks on cylinder head surfaces without the benefit of close proximity or adequate lighting conditions is a particularly difficult task. In addition, manually annotating large datasets is laborious and inefficient [14].

This paper presents a novel method for enhancing the ability of object detection models to identify cracks on cast iron surfaces when the amount of training data is limited. The primary innovation lies in our approach to synthetic data augmentation, which is explicitly tailored to improve the detectability of small cracks. Unlike conventional data augmentation techniques that often involve generic transformations, our method involves the strategic replication of cracks from one spatial location to another across images. This not only preserves the integrity of crack characteristics but also simulates varied realistic scenarios in which cracks might manifest under different conditions on similar surfaces. The incorporation of engineering knowledge further differentiates our method from existing techniques. By understanding where cyclic thermo-mechanical loads typically lead to cracks, we enhance the model's ability to predict crack locations more accurately, significantly improving the recall of potential defects. This is particularly crucial in industrial applications where prioritizing recall—detecting all possible cracks—is more important than reducing false positives (precision). Performance metrics such as the F2 score, which emphasizes recall, along with MAP, are utilized to evaluate the effectiveness of our method. Both metrics have shown substantial improvement over traditional augmentation techniques and generative adversarial networks (GANs), especially in terms of real-world applicability and computational efficiency. While GANs and other advanced AI techniques have proven effective in generating realistic data, their application to crack detection presents challenges. GANs require large, well-labeled datasets and significant computational resources, which are not always feasible in industrial settings where data may be scarce and real-time, cost-efficient solutions are needed. In contrast, our engineering-informed approach provides a simpler, scalable solution that integrates domain-specific knowledge, allowing for improved model training even with limited data.

A secondary aim of this study is to assess the impact of this engineering-informed synthetic data generation compared to alternative methods, such as randomly placing synthetic cracks in locations unlikely to experience mechanical stress. Our findings show that placing cracks in expected, high-stress areas significantly improve the model's performance. Controlled experiments validated this hypothesis, demonstrating that cracks placed based on mechanical stress patterns led to better detection rates and model accuracy than those placed randomly.

While sample pretreatment methods such as dye penetrant testing (e.g., DPT) can enhance crack visibility, this study focuses on synthetic augmentation techniques to provide a simple, scalable approach for environments where complex pretreatment procedures may not be feasible or practical. Moreover, we acknowledge that more advanced methods, such as GANs, could potentially outperform this approach in some scenarios; however, our method offers an accessible, computationally efficient alternative that can be implemented by engineers without extensive expertise or resources, making it more suitable for real-world industrial applications. Figure 1 provides an overview of the proposed method, illustrating the systematic integration of engineering knowledge into each phase of the data generation process. The subsequent sections of this paper will detail the literature review (Section 2) and the methodology employed (Section 3), present the results (Section 4), discuss the findings in a broader context (Section 5), and conclude with the study's limitations and potential future directions (Section 6).

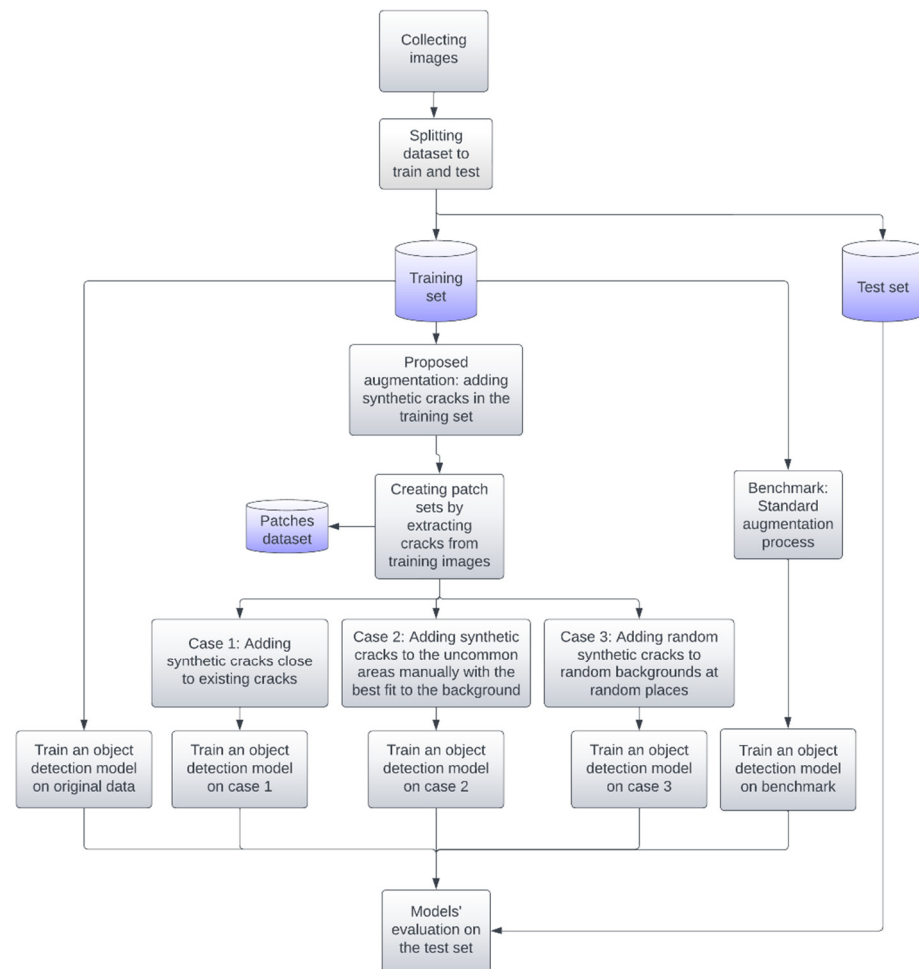


Figure 1. Overview of the proposed methods and evaluation procedures.

2. Literature Review

In engine component inspection, skilled machinists and mechanics regularly perform manual crack inspections using a combination of spray, magnifying glass, and magnetic particle inspection methods [15–17]. These methods are laborious and time-consuming, and their effectiveness is highly dependent on the subjective judgment and biases of the inspectors, which may result in inaccuracies and thus the omission of certain flaws [18]. Recently, there has been a growing interest in developing image-based automatic or semi-automatic defect detection and classification techniques to circumvent these limitations. These approaches employ conventional digital image processing methods, such as thresholding and segmentation-based and edge-detection techniques. Notably, Mohan et al. [19] present a comprehensive survey of various image processing-based strategies for crack detection. This emerging research area can potentially improve the efficiency and accuracy of engine component inspection, thereby overcoming the obstacles presented by manual inspection techniques. To address the concerns associated with manual inspection, semi-automated and automated inspection tools are utilized to aid in the inspection process. According to Mohan et al. [2], semi-automated systems, which involve both human inspectors and machine vision systems, offer benefits such as reduced inspection time, fewer errors (approximately 47 percent fewer), and reduced variation in defect detection.

Typical defect detection techniques include ultrasonic testing, osmosis testing, and X-ray analysis [20]. Similarly to X-ray testing, ultrasonic techniques are used to identify flaws within the internal structure of the test subject. These techniques utilize filtering to facilitate the extraction of features and the description of detected defects. In addition to these conventional techniques, diverse applications have witnessed the emergence of deep

learning-based defect detection methods in recent years. These algorithms frequently utilize deep neural networks, including CNNs, residual networks, and recurrent neural networks. Deep learning-based computer vision defect detection applications have demonstrated remarkable accuracy in detecting binary defects (present or not) [21]. Ren et al. [22] introduced an effective method to mitigate the negative impact of product defects by addressing the current state-of-the-art defect detection-based machine vision. On the other hand, Wang Liqun et al. [23] focused on defect detection using deep learning, specifically CNNs, to train and analyze large sets of image acquisition data. Their work demonstrates that this method can effectively extract features and achieve accurate and efficient defect classification.

CNNs have emerged as prominent architectures for object detection and image classification tasks in contemporary vision-based systems. Due to their training on large datasets such as GoogleNet and ImageNet, ConvNet models, such as VGGNet [24], InceptionV3 [25], AlexNet [26], ResNets [27], and GoogleNet [28], have garnered considerable attention and demonstrated exceptional performance. Recently, CNN-based algorithms have been applied to various defect detection problems in automotive engineering and related fields. These applications include precise part defect detection in automotive engines [29], the classification and detection of highway cracks [30], the detection and recognition of defects in welding images [30], online defect detection in weld images [31], and the detection of automotive paint defects [32]. The successful application of ConvNet-based methods in these contexts demonstrates the capability of these architectures to improve the precision and effectiveness of defect detection in real-world scenarios.

In addition to their application in general defect detection, CNN-based algorithms have also found extensive use in a more specialized domain: the detection of cracks. In the field of surface crack detection, numerous CNN-based models have been developed to detect cracks on various surfaces. These applications include the detection of cracks in nuclear power plant components [33], tunnel cracks [34,35], bridge cracks [36], cracks in concrete structures [37], pavement cracks [38], and roadway cracks [39]. In addition, recent developments have led to the incorporation of CNN-based neural networks for defect detection into autonomous robotic systems. These autonomous robotic platforms equipped with CNN-based models have effectively identified and quantified tunnel defects, providing valuable data for assessing tunnel stability [40–42]. This integration demonstrates the potential of combining deep learning techniques with robotics to improve the detection and monitoring of defects in critical structures. In the past five decades, digital image processing techniques have stimulated extensive research regarding crack detection technology [43,44]. Applications of this research include the detection of cracks in beams, bridges, pharmaceutical drugs, building structures [45], concrete [46], glass bottles, engineering materials, medical bones, pavements [47,48], road surfaces [49], subway tunnels, and walls. Diverse, innovative techniques for crack detection have been developed and implemented to address these diverse scenarios.

While CNNs have shown promise in accurately detecting cracks in structures, it is essential to recognize that the optimal use of deep neural networks is highly dependent on having a substantial amount of annotated training data. Training neural networks utilizing deep learning with insufficient data can result in overfitting and subpar predictive performance. When the quality and quantity of training data are increased, the effectiveness of deep neural networks increases significantly, enabling the models to generalize better and achieve higher performance levels [12].

Although numerous open-source crack datasets, such as those focusing on road surface cracks [13,50] and concrete surface cracks [51,52], are readily available, the lack of dedicated datasets specific to cylinder head surface cracks poses a significant barrier to the viability of cylinder head surface crack detection. Furthermore, the laborious and inefficient nature of manually labeling large datasets exacerbates the difficulty of obtaining suitable data for this specific application.

Multiple data augmentation techniques, such as rotation, inversion, and scaling, have been intensively studied to enlarge datasets in computer vision applications, such as

classification [26], object detection [53], and instance segmentation [54]. These techniques are intended to improve the generalization and performance of deep learning models by increasing the diversity of training data. Despite the application of image transformations for data enhancement, it is essential to recognize that the inherent disparity between cylinder head surface crack data and the crack data of other surfaces remains substantial and cannot be effectively reduced by these image transformation approaches alone [55].

Nevertheless, applying data augmentation via synthetic images provides a promising avenue for introducing greater diversity and expanding the dataset. GANs have emerged as an effective method for augmenting data with synthetic images. GANs have demonstrated the ability to generate realistic, high-quality images without labels, establishing them as a viable method for generating synthetic data [56–58]. Deep learning GANs have been used to generate synthetic medical datasets in the medical domain, resulting in improved CNN performance for medical image classification tasks [58]. In addition, a simple method for generating synthetic datasets involving the insertion of object masks from authentic images has been proposed as a practical alternative to elaborate graphic renderings and complex scene composition, simplifying implementation without sacrificing photorealism [14].

To the best of our knowledge, the application of GANs to generate synthetic datasets for crack detection on metal surfaces remains largely unexplored within the domains of computer vision and defect analysis. While GANs have proven effective in generating synthetic data for tasks such as medical imaging, their computational intensity, large data requirements, and the complexity of tuning them for specific applications, such as crack detection on metal surfaces, pose significant challenges in industrial settings where real-time, cost-efficient solutions are required. Additionally, GANs demand expertise in hyperparameter tuning and access to substantial computational resources, making their implementation less feasible in many industrial environments. Although synthetic datasets have been extensively used in various fields, their application to detecting small defects—particularly cracks in cylinder heads—has been limited. In this study, we address these limitations by introducing an engineering-informed synthetic data augmentation method that strategically replicates cracks based on mechanical stress patterns unique to cylinder head surfaces. This approach differs from relying on computationally expensive methods such as GANs by offering a practical, scalable solution to enhancing crack detection in industrial applications. Our method leverages existing crack data to generate new, contextually relevant training examples, thereby improving the model's ability to detect cracks while maintaining computational efficiency. By tailoring our augmentation technique to address the specific needs of industrial quality inspection, particularly for small, hard-to-detect cracks, we provide a solution that can be implemented with limited datasets and fewer resources than methods like GANs. This approach is aligned with the requirements of real-world industrial environments, where time and computational efficiency are critical. Our study not only bridges the current data gap but also establishes a more accessible, effective foundation for crack detection systems, ensuring that industries can adopt advanced artificial intelligence methods without the need for extensive computational infrastructure or deep technical expertise.

3. Methodology

3.1. Dataset Preparation

This work utilized images of cylinder heads from John Deere Reman. As shown in Figure 2, we used two sets of images obtained by two distinct methods for this investigation. The first, more controlled approach at Iowa State utilizes real cylinder heads with cracks provided by John Deere Reman and is conducted in a laboratory setting. The second, less controlled approach in a remanufacturing shop environment uses a simple gantry system to acquire images using a conventional, integrated lens SLR camera with a resolution of 2592 by 1944 pixels, with less control than the laboratory environment, affecting the image quality. This investigation required this second method to acquire additional defect images to aid model training. These images, in ambient light, captured in two distinct

environments, are combined to form a single, curated image dataset. Overall, 12% of the images were obtained using the lab setting, and the rest were obtained using the less controlled method. We randomly divided our dataset into training and test sets, keeping the percentage of images for each method the same in both the training and test sets. A total of 91 crack images with 113 cracks in the image set were used for training, while 68 crack images with 89 cracks were used for testing. “Labeling” was used to annotate the dataset.

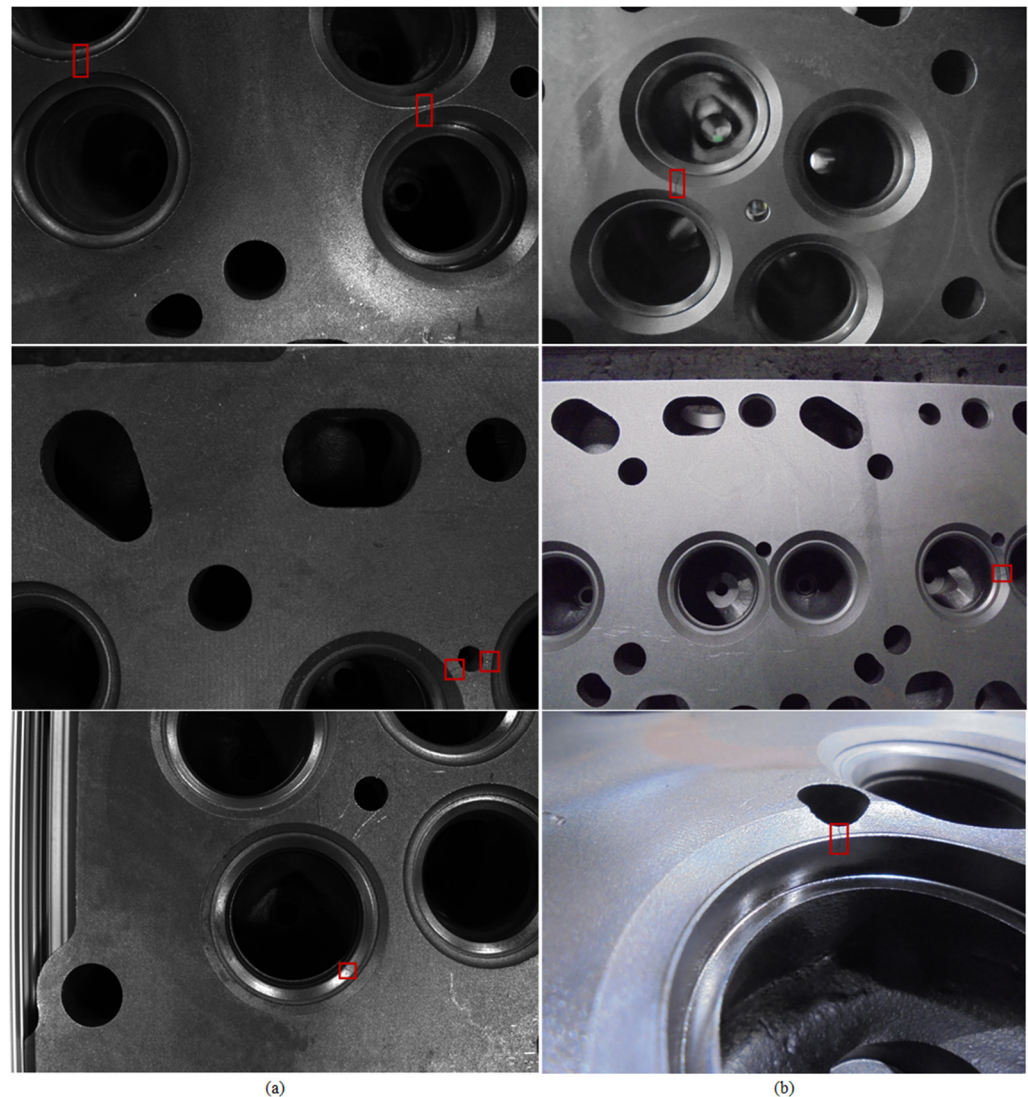


Figure 2. Sample images and cracks (small red squares) obtained in different methods: (a) lab setting and (b) less controlled setting.

3.2. Model Evaluation

In image detection, MAP is widely used to evaluate similar modes. In fact, this is the core evaluation metric used by the YOLO algorithm employed here. However, for this application, recall is critical; missing a crack is a more severe error than identifying a non-crack incorrectly. A weighted measure of precision and recall is commonly obtained using the harmonic mean, namely the F-score. An equal weight is achieved with the F1 score, and for higher values more weight is given to recall. We chose to double the weight of recall and use the F2 score as we believe it is appropriate for this application. On the other hand, the MAP metric has become standard in object detection models, and YOLO algorithms use this as a primary metric. The MAP metric seeks to achieve a balance between recall and precision, making it a valuable evaluation tool. For our method, it is intuitively likely that adding synthetic cracks, especially in areas where cracks should not occur, may

decrease precision. Reporting the MAP metric is useful because, although improving the F2 score is the primary goal, it is desirable to at least maintain or possibly improve the overall MAP metric. Using both metrics enables us to confirm that the proposed method effectively improves recall while maintaining or enhancing the overall balance between recall and precision.

For a mathematical motivation of the F2 score, note that the model's predictions were compared against the ground truth annotations to determine true positives (*TPs*), true negatives (*TNs*), false positives (*FPs*), and false negatives (*FNs*). The precision and recall are mathematically defined as

$$\text{Precision} = \frac{TP}{(TP + FP)},$$

$$\text{Recall} = \frac{TP}{(TP + FN)}.$$

Given the above definitions, the $F\beta$ -score is defined as a weighted harmonic mean (equal weight if $\beta = 1$):

$$F\beta = (1 + \beta^2) \times \frac{\text{precision} \times \text{recall}}{(\beta^2 \times \text{precision}) + \text{recall}}.$$

And we notice that with $\beta = 2$ the weight for recall is higher.

3.3. Patching Process and Augmentation

In this section, we describe the approaches taken to enhance object detection performance in scenarios characterized by a limited dataset containing low-quality images, blurriness, and varying lighting conditions. Our research examines various approaches to increase the number of defect instances and the size of the training set, then evaluates their impact on the model's performance. Noting that the test set remained unchanged throughout this process ensures a consistent and reliable evaluation of the model's performance.

3.3.1. Standard Data Augmentation

As a benchmark to the new approach, several standard augmentation techniques were employed to expand the training set and increase the variability of our dataset. These methods encompassed operations such as horizontal and vertical flipping, transposing, center cropping, and all combinations thereof. Figure 3 offers a selection of sample images that result from these transformations. By applying these transformations, we could generate new images with varying orientations, perspectives, and sizes. Each image in the original training set was augmented 10 times using the abovementioned techniques. Consequently, the initial training set, consisting of a few images, was greatly augmented. Ultimately, this augmentation procedure produced a training set containing 1001 images with 1243 cracks. These standard augmentation techniques were intended to introduce additional variations and increase the diversity of the training set. By doing so, we aimed to improve the model's ability to generalize and detect more cracks under different orientations, perspectives, and scales.

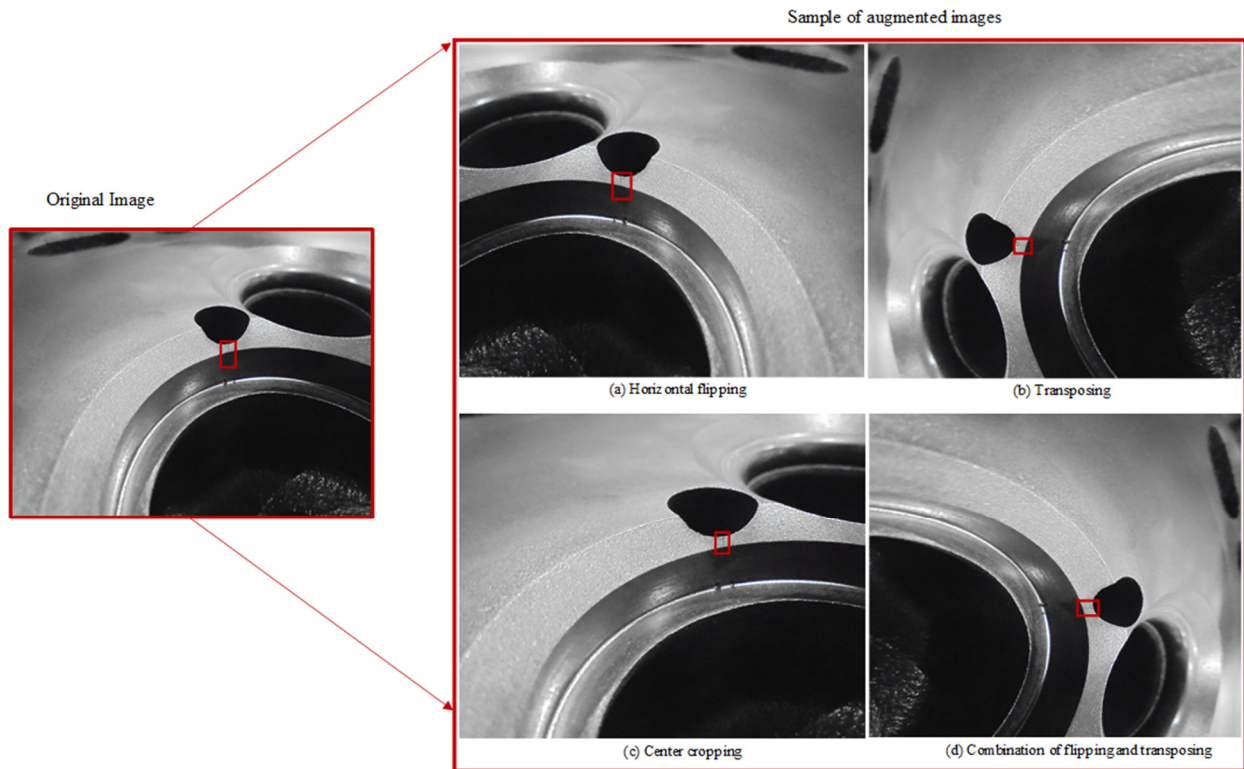


Figure 3. Sample cracks (small red squares) in original image vs. augmented images.

3.3.2. Data Augmentation by Adding New Cracks in Potential Locations

This section presents the idea of augmenting the training set by incorporating synthetic cracks into the cylinder images. We propose a method that utilizes crack patches extracted from the training set (Figure 4) to generate additional cracks in the expected locations, given that JD cylinder head cracks are expected to appear in specific locations (Figure 5). To accomplish this, we created a separate patch dataset containing all of the training set's cracks. These patches serve as the foundation from which new cracks are generated. We carefully select patches that align with the image's background color in the training set and seamlessly pasted them into regions where cracks are expected to be observed (Figure 6). Following this methodology, we created new datasets incorporating these patched cracks. This procedure improves the diversity of crack instances in the training data. This experiment examines the effect of augmenting the training set with synthetic cracks in areas where cracks are common. By incorporating these additional instances, we hoped to improve the model's ability to detect cracks in the targeted areas.

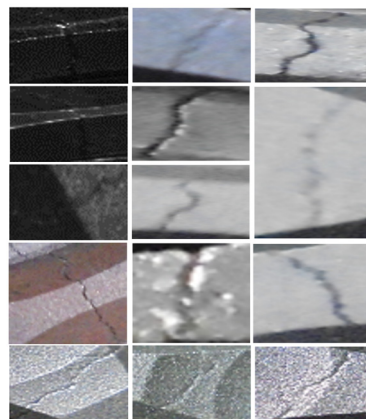


Figure 4. Sample crack patches extracted from training set.

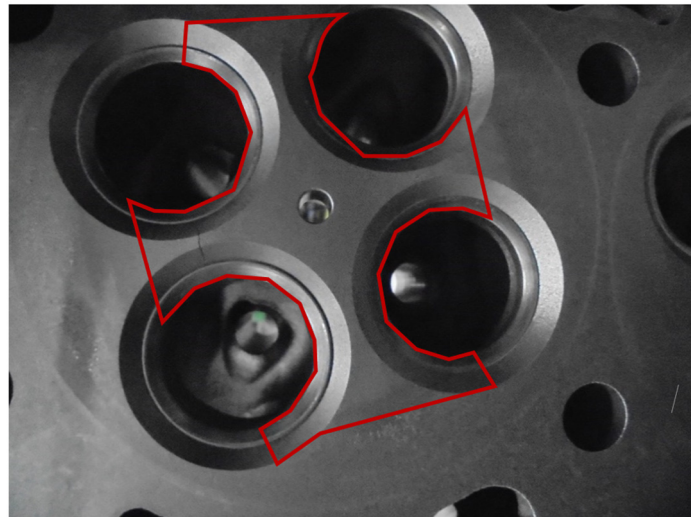


Figure 5. Potential (known) areas where cracks may occur.

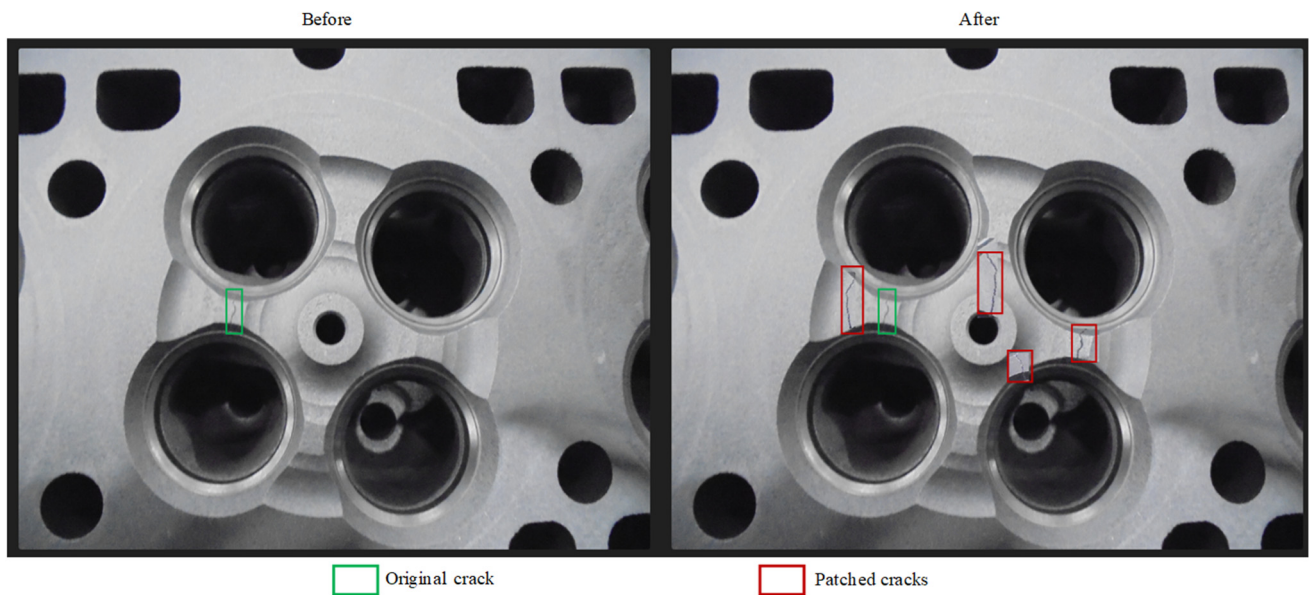


Figure 6. Sample of an image before and after adding synthetic cracks to known areas.

3.3.3. Data Augmentation by Adding New Cracks in Uncommon Locations

This section describes the process of augmenting the training set with synthetic cracks in areas of the cylinder heads where cracks are not typically observed. In this method, we manually extracted cracks from the patches dataset and pasted them onto training set images. The cracks were added to the locations where they usually do not occur. These synthetic cracks were placed meticulously to ensure consistency with each image's background (Figure 7). Via this method, we hoped to comprehensively evaluate the effect of placing synthetic cracks in unanticipated regions and compare it to the effect of placing cracks in expected regions. This analysis provides valuable insights into the model's performance and ability to detect cracks in expected and unexpected areas.

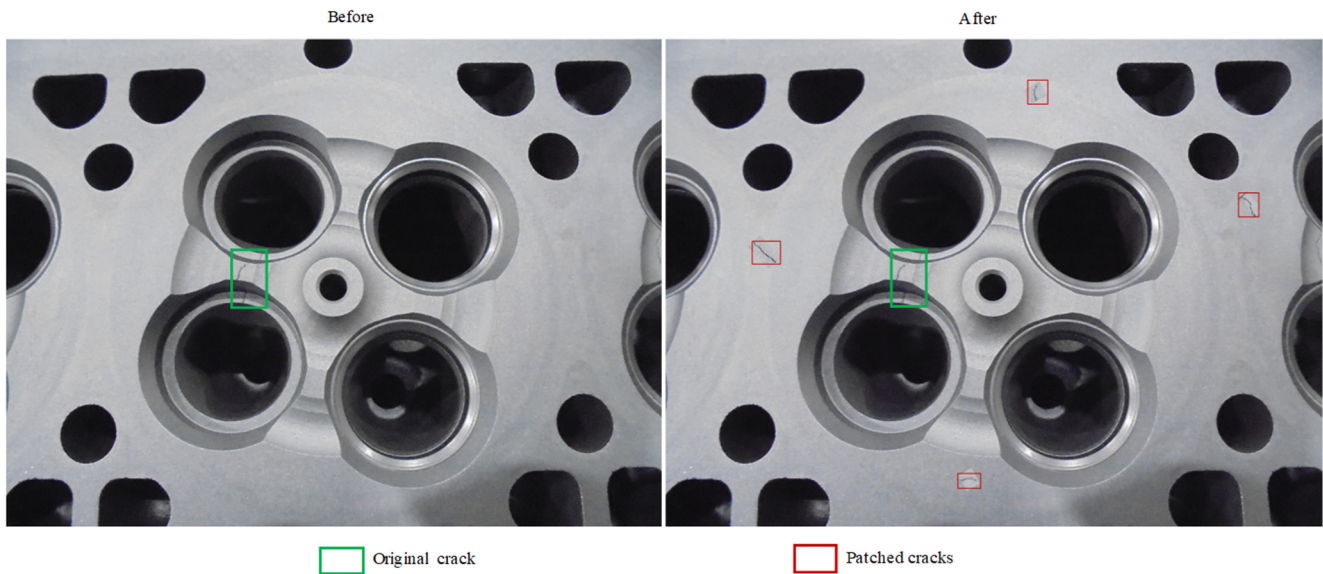


Figure 7. Sample of an image before and after adding synthetic cracks to uncommon areas.

3.3.4. Data Augmentation by Randomly Adding Synthetic Cracks

This section elucidates how crack patches were systematically dispersed among the training images through a randomization procedure. We initially selected a random patch and a background image from our training set to achieve this. Subsequently, as shown in Figure 8, the chosen patch was placed randomly on the surface of the cylinder head. This random placement of cracks on the surfaces of training images was implemented to assess the impact of synthetic crack positioning.

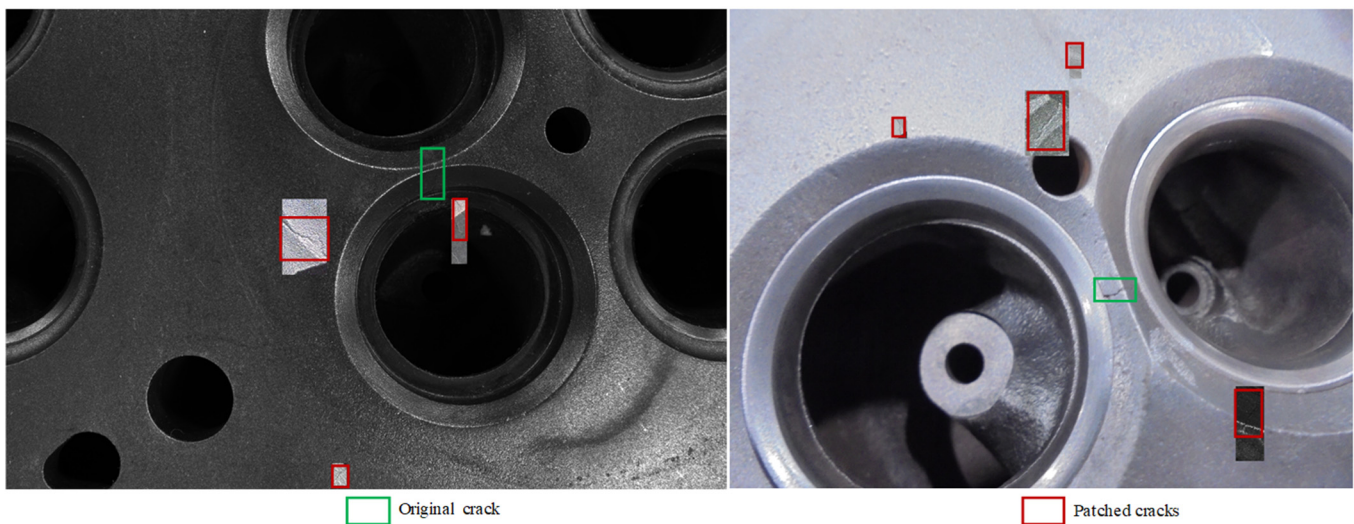


Figure 8. Sample of training images after randomly placing synthetic cracks.

3.4. Model Architecture Selection and Training

We annotated the dataset to train the model for crack detection by manually labeling the crack regions in the images. Experienced domain experts carefully marked the crack boundaries using bounding boxes. This ground truth annotation served as the reference for training and evaluating the crack detection model. We explored different object detection architectures to identify the most suitable model for crack detection on cylinder heads. Considering the complexity and variability of crack patterns, we compared two prominent architectures, YOLO v8x and Faster RCNN Resnet 101, based on their performance on the COCO dataset, a widely used benchmark for object detection models. We relied

on the existing literature and research that had reported results on the COCO dataset when evaluating the performance of both architectures. Across multiple object detection tasks, YOLOv8x demonstrated superior accuracy and faster training, and prediction times compared to Faster RCNN Resnet 101. Based on these findings from the COCO dataset, we chose YOLOv8x as the model architecture for our study. The selected object detection model was trained using a subset of the annotated dataset. We divided the dataset into training, validation, and test sets, ensuring that images from the same cylinder head were present in only one of these sets to avoid data leakage.

3.5. Experimental Setup

All experiments were conducted on a high-performance computing system equipped with powerful GPUs to accelerate the training and evaluation processes. We utilized deep learning frameworks like TensorFlow to implement and train the crack detection models. In all experiments conducted during this study, we ensured that the same hyperparameters were consistently applied across different configurations of the object detection models. Key hyperparameters, such as learning rate, batch size, optimizer settings, and the number of training epochs, were kept constant to ensure that any observed variations in model performance were solely due to the augmentation techniques being evaluated. This controlled setup enabled us to rigorously assess the impact of each augmentation method, ensuring that improvements in metrics such as recall and MAP could be directly attributed to the synthetic data augmentation process, without confounding effects from varying hyperparameters.

4. Results

In this section, we present the results of our crack detection experiments. The main results compare the performance of YOLOv8x trained on the original training set versus employing various augmentation techniques, including standard augmentation, adding cracks to expected areas, adding cracks to unexpected areas, and adding random cracks to the training set. To evaluate the effect of the model, we also report a limited comparison of YOLOv8x and Faster RCNN Resnet 101. For all the results, we assess the performance using two metrics: F2 score and MAP0.5. Consistent hyperparameters were used across all experiments to ensure that any observed differences in model performance were attributable solely to variations in the training set; that is, the original data versus the augmented data.

The original training data contains 91 crack images with 113 cracks in the image set, and the test data contain 68 crack images with 89 cracks. Each model was trained for 100 epochs. We then incremented the number of cracks up to 400 on the training set by adding synthetic cracks—first: to the places where we expect to see cracks, second: to the places where we do not expect to see cracks, and third: to the random places on the cylinder head—and then evaluate the models' performance. In all experiments, the test set is unchanged for every evaluation.

Table 1 shows the results. The reported findings demonstrate that incorporating synthetic cracks into expected areas has the best performance. Compared to using the original data directly, this type of data augmentation yields notable improvements in the model's F2 score, by approximately 62%, and improves the MAP score by approximately 40%. Furthermore, the results indicate that introducing synthetic cracks randomly or including unexpected areas boosts the F2 score by approximately 38%, while the MAP score remains relatively unchanged. Nevertheless, this method is less effective than adding synthetic cracks to the expected regions. On the other hand, using standard image augmentation techniques, such as cropping, rotating, and flipping, leads to an enhancement of the test set MAP by approximately 25%. However, these techniques cause a decline in the model's F2 score by 10%. Recall that for standard augmentation we applied augmentation techniques to each image in the training set 10 times, expanding the dataset from 91 to 1001 images with 1243 cracks, which makes it a significantly larger training dataset than when adding synthetic cracks.

Table 1. Results of adding synthetic cracks to different places.

		Number of Training Cracks	Crack Placement for Synthetic Cracks					
			Expected Areas		Both Expected and Unexpected Areas		Random Areas	
			F2	MAP	F2	MAP	F2	MAP
Benchmarks	Original	113	0.43	0.40	0.43	0.40	0.43	0.40
	Standard augmentation	1243	0.39	0.50	0.39	0.50	0.39	0.50
Synthetic cracks	50	163	0.49	0.45	0.53	0.43	0.51	0.44
	100	213	0.54	0.35	0.55	0.40	0.50	0.40
	150	263	0.51	0.46	0.57	0.43	0.65	0.40
	200	313	0.59	0.48	0.58	0.43	0.58	0.36
	300	413	0.68	0.56	0.58	0.40	0.55	0.42
	350	463	0.70	0.57	0.55	0.38	0.56	0.40
	400	563	0.70	0.56	0.58	0.39	0.50	0.36
350 synthetic cracks + standard augmentation		4630	0.69	0.55	0.55	0.47	0.52	0.49

As the traditional augmentation methods primarily improve MAP score and the new approach improves F2 score more, there is reason to believe that combining the two would improve both results. We test this by applying the previously discussed standard augmentation techniques to the dataset containing 350 synthetic cracks, which appears to be the point at which adding more cracks does not improve performance further. The results of this evaluation are shown in the last row of Table 1 and indicate that while integrating these two methodologies leads to enhancements in MAP when including unexpected or random areas, it falls short of surpassing the outcomes achieved by simply incorporating 350 synthetic cracks into expected areas utilizing engineering knowledge. In summary, the results suggest that augmenting the cylinder head surface with synthetic cracks, based on engineering knowledge regarding plausible crack locations, results in the most effective approach among the various augmentation techniques tested.

Many of the improvements in measured performance appear substantial, but for added rigor, we also estimate the variability to determine if they are statistically significant. To obtain an estimate of the variance, we ran eight replications by adding 100 cracks randomly (see the third row of Table 1 above for a single replication). We carried this out for the random placement version to reduce any bias due to manual placement. For the F2 score, the range of values was 0.50–0.68 with a mean of 0.59, a standard deviation of 0.06, and a 95% t-confidence interval of [0.54,0.65]. For the MAP measure, the range of values was 0.37 to 0.45, with a mean of 0.41 and 0.04 and a 95% t-confidence interval of [0.38,0.44]. There is, thus, significant variability in the results. Still, the increase in the F2 score may be considered significant, while the MAP value does not significantly change for the random placement of cracks.

For the final numerical results, we assess the influence of introducing synthetic cracks to designated regions on a different object detection model. We employed Faster RCNN Resnet 101 for training on the dataset. The results show that the Faster RCNN Resnet 101 with the original dataset can achieve an F2 score and MAP of 0.40 and 0.32, respectively. After adding 300 synthetic cracks at potential areas, the F2 score and MAP improved by 47.5% and 34.3%, respectively. These results indicate that adding synthetic cracks can offer advantages for other object detection models.

Apart from the quantitative improvements previously presented, it is of interest to determine some intuition regarding how the crack detection is improved. To that end,

Figure 9 visually depicts the outcomes of YOLOv8x in terms of predicted images before and after integrating 400 synthetic cracks into specific regions. Including synthetic cracks within the model's training data led to identifying cracks that had previously gone undetected, a capability absent in the model that did not incorporate synthetic cracks. These newly recognized cracks are of a type that had relatively low representation within the original training dataset. Through the augmentation of the dataset with these specific crack types, the model demonstrated an improved proficiency in effectively identifying them.

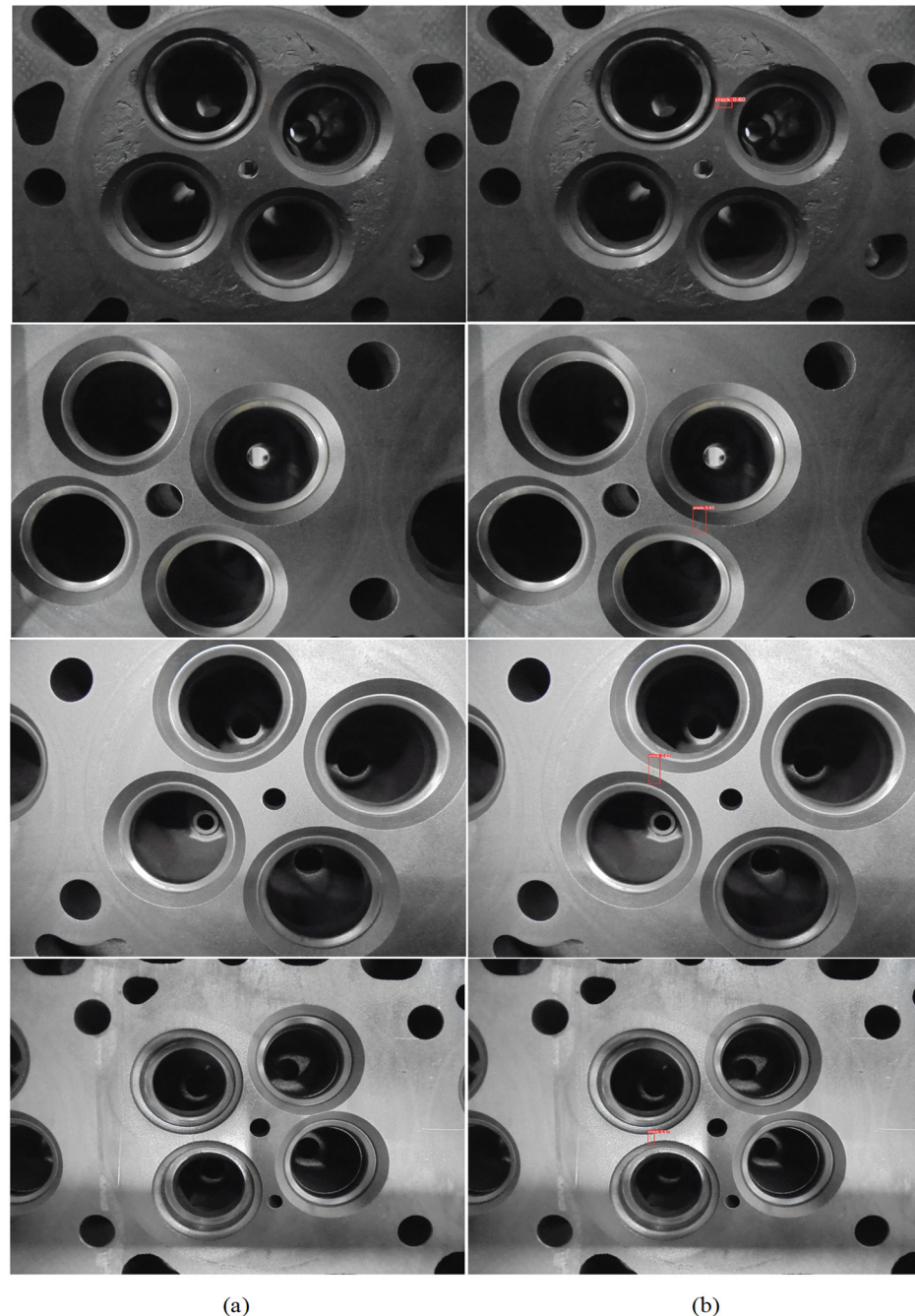


Figure 9. Sample of predicted images using YOLOv8x before (a) and after (b) adding 300 synthetic cracks.

5. Discussion

The findings presented in the previous section provide evidence that strategically incorporating engineering knowledge by adding synthetic cracks at locations where cracks are likely to occur is the most advantageous method for augmenting the training set,

resulting in simultaneous improvements to the F2 and MAP metrics. This technique is superior to randomly placing or positioning synthetic cracks in uncommon regions and standard data augmentation techniques. Intuitively, this approach prevents confusion as adding synthetic cracks in regions where cracks are not expected to exist may cause false positives in the model; that is, the prediction of cracks where there is no crack. Additionally, this method is more efficient because efforts are made to align the destination background with the new crack background, preserving visual coherence during augmentation. When synthetic cracks were randomly introduced or placed in uncommon locations, the model's F2 score improved, but the effect on the MAP metric was insignificant. This phenomenon is attributable to the introduction of new regions where cracks do not usually form. Even though these synthetic cracks mimic crack-like patterns, their placement does not align with the actual distribution of cracks in the dataset, resulting in a marginal improvement in the overall detection accuracy.

Further motivating the use of the proposed data augmentation approach, the results show that standard augmentation techniques, such as cropping, flipping, and others, did not significantly improve the performance of crack detection. Numerous images containing a single crack added to the dataset may have resulted in an excessive influx of irrelevant information, making it difficult for the model to learn and generalize from the augmented dataset. In addition, the synthetic cracks generated by standard augmentation techniques exhibited a high degree of similarity to the original cracks in their respective locations, providing the model with limited variation and potentially hindering its ability to generalize and detect cracks in new regions or orientations effectively.

The standard augmentation techniques primarily result in the enhancement of MAP, whereas the introduction of synthetic cracks in various regions primarily improves the F2 score. The combination of these two approaches improves the MAP compared to adding synthetic cracks in unexpected or random areas. However, this combined approach yields similar MAP results as those achieved when incorporating only synthetic cracks in expected areas. These findings underscore the continued efficacy of integrating engineering knowledge into the process of introducing synthetic cracks as the most effective means of enhancing the model's performance.

Moreover, the results indicate that cracks exhibiting characteristics closely aligned with most of the training set in terms of size and background, despite some images of suboptimal quality, were successfully detected upon adding synthetic cracks that accurately captured these characteristics. Despite introducing synthetic cracks, certain cracks in our dataset, particularly those located at the image corners or subjected to excessive zoom, resulting in significantly larger crack sizes compared to the regular cracks in the dataset, remained undetected. Future work will aim to introduce more synthetic cracks representing these outliers to improve the model's ability to detect uncommon crack types. The results of this study indicate that the introduction of synthetic cracks via the replication of cracks from the training set and their strategic placement in regions where cracks are anticipated leads to significant improvements in the model's F2 score, emphasizing the importance of crack detection (recall), as well as improvements in MAP. This straightforward yet efficient method proves especially useful in situations characterized by few available data points (cracks) and significant variation in their presentation. For instance, in our study, the cracks observed in various images exhibit a wide range of variations in terms of lighting conditions, distances from the surface, and other factors that may hinder the proper training of the model. The proposed synthetic crack augmentation technique significantly improves the model's performance, mitigating the challenges posed by limited and diverse data instances and enabling more accurate and robust crack detection capabilities.

While the results of this study emphasize the importance of recall in crack detection for industrial applications, we recognize that increasing recall through the augmentation of synthetic cracks may lead to a rise in false positives, thereby impacting precision. In particular, we observed that when synthetic cracks were placed in locations less likely to experience mechanical stress, the model occasionally detected cracks in regions where none

existed, leading to a higher rate of false alarms. This trade-off between recall and precision is an inherent challenge in defect detection, where the primary goal is to avoid missing potential defects, even at the cost of misidentifying non-cracks. However, when cracks were strategically placed in areas informed by engineering knowledge, the model maintained a more favorable balance between recall and precision, resulting in minimal impact on precision while substantially improving recall. Future work will aim to address this trade-off by refining the placement of synthetic cracks and introducing more sophisticated techniques to reduce false positives, thus maintaining high recall without significantly sacrificing precision.

Finally, the findings presented in the preceding section suggest that the crack detection-specific approach employed for generating synthetic data holds potential benefits not solely confined to YOLO algorithms. This approach also exhibits the capability to enhance the performance of alternative object detection models, including Faster R-CNN. The application of our method to multiple architectures demonstrates its generalizability and practical relevance in a wide range of industrial applications where detecting small defects is critical, including Faster R-CNN.

6. Conclusions and Future Work

In our research, we encountered limitations concerning the availability of real-world cylinder head images with cracks for training data. Despite this constraint, we generated synthetic cracks and evaluated our proposed method on a validation set comprising 68 images containing 87 cracks. The results demonstrated that the engineering-informed synthetic data augmentation approach significantly improved the model's performance, particularly in detecting small, hard-to-detect cracks, while being computationally efficient. However, to further strengthen the generalizability and robustness of our approach, future work should aim to collect a more extensive and diverse dataset of cylinder heads with cracks. A larger dataset would enable us to evaluate the method on a broader range of test cases and improve the reliability of the results.

While we recognize that combining sample pretreatment methods, such as solvent-based DPT, with the proposed synthetic augmentation techniques could potentially enhance detection performance further, it was beyond the scope of this paper. The primary objective was to demonstrate the effectiveness of a simpler, scalable approach that does not rely on complex sample pretreatments, making it more accessible for industrial applications where data and computational resources are limited. However, we acknowledge that incorporating pretreatment techniques in future research could provide additional insights into how these methods can complement synthetic data augmentation, potentially leading to further improvements in model accuracy.

Another area worth exploring is refining the synthetic crack generation process. In this study, we adopted a straightforward approach by copying crack patches from the training set and pasting them into other regions of the images. While this method effectively improved model performance, it did not fully adapt the patch background to match the image background. In future work, we plan to enhance the realism of the synthetic cracks by fitting the patch background more seamlessly with the destination image background. This refinement is expected to result in more realistic synthetic cracks and, consequently, better performance in crack detection tasks.

In addition to cylinder head cracks, we plan to expand our dataset to include cracks from other domains, such as concrete cracks, to assess the transferability of the proposed method. By incorporating diverse crack types into our training data, we aim to explore how well the model can generalize across different domains. This will provide valuable insights into the model's robustness and adaptability, potentially leading to the development of a more versatile crack detection system capable of handling a variety of crack types in different industrial contexts.

In conclusion, as part of our future research, we will continue to address the limitations of our current approach by collecting a more extensive dataset of cylinder head images

with cracks, refining the synthetic crack generation process, and incorporating sample pretreatment methods like DPT. These efforts will contribute to the ongoing development and improvement of our crack detection methodology, making it more effective and applicable in real-world scenarios, while ensuring that it remains practical and scalable for industrial use.

Author Contributions: Conceptualization, M.M., S.O. and G.E.O.K.; methodology, M.M., S.O. and G.E.O.K.; software, M.M.; validation, M.M., S.O. and G.E.O.K.; formal analysis, M.M., S.O. and G.E.O.K.; investigation, M.M., S.O. and G.E.O.K.; resources, M.M., S.O. and G.E.O.K., P.A.K.; data curation, M.M., P.A.K.; writing—original draft preparation, M.M., S.O. and G.E.O.K.; writing—review and editing, M.M., S.O. and G.E.O.K.; visualization, M.M.; supervision, S.O. and G.E.O.K.; project administration, S.O. and G.E.O.K.; funding acquisition, G.E.O.K. and P.A.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by U.S. Department of Energy’s Office of Energy Efficiency and Renewable Energy (EERE): grant number DEEE0007897–RM05.

Data Availability Statement: The data presented in this study are available on request from the corresponding author due to company privacy.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Paterson, D.A.P.; Ijomah, W.L.; Windmill, J.F.C. End-of-life decision tool with emphasis on remanufacturing. *J. Clean. Prod.* **2017**, *148*, 653–664. [CrossRef]
2. Mital, A.; Govindaraju, M.; Subramani, B. A comparison between manual and hybrid methods in parts inspection. *Integr. Manuf. Syst.* **1998**, *9*, 344–349. [CrossRef]
3. Errington, M.; Childe, S.J. A business process model of inspection in remanufacturing. *J. Remanuf.* **2013**, *3*, 7. [CrossRef]
4. Hammond, R.; Amezcuita, T.; Bras, B. Issues in the Automotive Parts Remanufacturing Industry: Discussion of Results from Surveys Performed among Remanufacturers. *Eng. Des. Autom.* **1998**, *4*, 27–46.
5. Kujawińska, A.; Vogt, K. Human Factors in Visual Quality Control. *Manag. Prod. Eng. Rev.* **2015**, *6*, 25–31. [CrossRef]
6. Drury, C.G.; Karwan, M.H.; Vanderwarker, D.R. The Two-Inspector Problem. *IIE Trans.* **1986**, *18*, 174–181. [CrossRef]
7. Cha, Y.-J.; Choi, W.; Büyüköztürk, O. Deep Learning-Based Crack Damage Detection Using Convolutional Neural Networks. *Comput.-Aided Civ. Infrastruct. Eng.* **2017**, *32*, 361–378. [CrossRef]
8. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [CrossRef]
9. He, K.; Zhang, X.; Ren, S.; Sun, J. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 1904–1916. [CrossRef]
10. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. 2016. pp. 779–788. Available online: <http://pjreddie.com/yolo/> (accessed on 15 August 2023).
11. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. April 2017. Available online: <https://arxiv.org/abs/1704.04861v1> (accessed on 15 August 2023).
12. Sabiri, B.; Asri, B.E.L.; Rhanoui, M. Efficient Deep Neural Network Training Techniques for Overfitting Avoidance. In *Enterprise Information Systems*; Springer: Cham, Switzerland, 2023; pp. 198–221. [CrossRef]
13. Shi, Y.; Cui, L.; Qi, Z.; Meng, F.; Chen, Z. Automatic Road Crack Detection Using Random Structured Forests. *IEEE Trans. Intell. Transp. Syst.* **2016**, *17*, 3434–3445. [CrossRef]
14. Dwibedi, D.; Misra, I.; Hebert, M. Cut, Paste and Learn: Surprisingly Easy Synthesis for Instance Detection. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017.
15. Mohammadzadeh, M.; Dubey, P.; Gunay, E.E.; Jackman, J.K.; Okudan Kremer, G.E.; Kremer, P.A. Deep Learning for Defect Detection in Inspection. In *Technology Innovation for the Circular Economy: Recycling, Remanufacturing, Design, Systems Analysis and Logistics*; Wiley: Hoboken, NJ, USA, 2024; pp. 143–155.
16. Asai, K.; Takeuchi, A.; Ueda, N.; Kawamoto, J. Computerized ultrasonic inspection system for ceramic pre-combustion chambers of automotive diesel engines. *SAE Trans.* **1985**, *94*, 959–969.
17. Almubarak, H.A.; Albloushi, A.H. Automotive Engine Tests “The Basics”. *Int. J. Eng. Res. Appl.* **2021**, *11*, 22–28.
18. Berwo, M.A.; Fang, Y.; Mahmood, J.; Yang, N.; Liu, Z.; Li, Y. FAECDD-CNet: Fast Automotive Engine Components Crack Detection and Classification Using ConvNet on Images. *Appl. Sci.* **2022**, *12*, 9713. [CrossRef]
19. Mohan, A.; Poobal, S. Crack detection using image processing: A critical review and analysis. *Alex. Eng. J.* **2018**, *57*, 787–798. [CrossRef]
20. Gholizadeh, S.; Leman, Z.; Baharudin, B.T.H.T. A review of the application of acoustic emission technique in engineering. *Struct. Eng. Mech.* **2015**, *54*, 1075–1095. [CrossRef]

21. Plessis, A.D.; le Roux, S.G. Standardized X-ray tomography testing of additively manufactured parts: A round robin test. *Addit. Manuf.* **2018**, *24*, 125–136. [[CrossRef](#)]
22. Ren, Z.; Fang, F.; Yan, N.; Wu, Y. State of the Art in Defect Detection Based on Machine Vision. *Int. J. Precis. Eng. Manuf.-Green Technol.* **2022**, *9*, 661–691. [[CrossRef](#)]
23. Wang, L.; Wu, J.; Wu, D. Research on Vehicle Parts Defect Detection Based on Deep Learning. *J. Phys. Conf. Ser.* **2020**, *1437*, 012004. [[CrossRef](#)]
24. Wu, J.; Zhang, Q.; Xu, G. Tiny Imagenet Challenge. *Technical Report*. 2017. Available online: <https://cs231n.stanford.edu/reports/2017/pdfs/930.pdf> (accessed on 16 September 2024).
25. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the Inception Architecture for Computer Vision. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 2818–2826.
26. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems*; Pereira, F., Burges, C.J., Bottou, L., Weinberger, K.Q., Eds.; Curran Associates, Inc.: Sydney, NSW, Australia, 2012. Available online: https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf (accessed on 4 April 2024).
27. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016.
28. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going Deeper with Convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015.
29. Qu, Z.; Shen, J.; Li, R.; Liu, J.; Guan, Q. PartsNet: A Unified Deep Network for Automotive Engine Precision Parts Defect Detection. In Proceedings of the 2018 2nd International Conference on Computer Science and Artificial Intelligence, Shenzhen, China, 8–10 December 2018; ACM: New York, NY, USA, 2018; pp. 594–599. [[CrossRef](#)]
30. Elghaish, F.; Talebi, S.; Abdellatef, E.; Matarneh, S.T.; Hosseini, M.R.; Wu, S.; Mayouf, M.; Hajirasouli, A.; Nguyen, T.-Q. Developing a new deep learning CNN model to detect and classify highway cracks. *J. Eng. Des. Technol.* **2022**, *20*, 993–1014. [[CrossRef](#)]
31. Zhang, Z.; Wen, G.; Chen, S. Weld image deep learning-based on-line defects detection using convolutional neural networks for Al alloy in robotic arc welding. *J. Manuf. Process* **2019**, *45*, 208–216. [[CrossRef](#)]
32. Akhtar, S.; Tandiya, A.; Moussa, M.; Tarry, C. An Efficient Automotive Paint Defect Detection System. *Adv. Sci. Technol. Eng. Syst. J.* **2019**, *4*, 171–182. [[CrossRef](#)]
33. Chen, F.-C.; Jahanshahi, M.R. NB-CNN: Deep Learning-Based Crack Detection Using Convolutional Neural Network and Naïve Bayes Data Fusion. *IEEE Trans. Ind. Electron.* **2018**, *65*, 4392–4400. [[CrossRef](#)]
34. Protopapadakis, E.; Voulodimos, A.; Doulamis, A.; Doulamis, N.; Stathaki, T. Automatic crack detection for tunnel inspection using deep learning and heuristic image post-processing. *Appl. Intell.* **2019**, *49*, 2793–2806. [[CrossRef](#)]
35. Ren, Y.; Huang, J.; Hong, Z.; Lu, W.; Yin, J.; Zou, L.; Shen, X. Image-based concrete crack detection in tunnels using deep fully convolutional networks. *Constr. Build. Mater.* **2020**, *234*, 117367. [[CrossRef](#)]
36. Zhang, C.; Chang, C.; Jamshidi, M. Concrete bridge surface damage detection using a single-stage detector. *Comput.-Aided Civil. Infrastruct. Eng.* **2020**, *35*, 389–409. [[CrossRef](#)]
37. Kim, B.; Yuvaraj, N.; Preethaa, K.R.S.; Pandian, R.A. Surface crack detection using deep learning with shallow CNN architecture for enhanced computation. *Neural Comput. Appl.* **2021**, *33*, 9289–9305. [[CrossRef](#)]
38. Ma, D.; Fang, H.; Wang, N.; Xue, B.; Dong, J.; Wang, F. A real-time crack detection algorithm for pavement based on CNN with multiple feature layers. *Road Mater. Pavement Des.* **2022**, *23*, 2115–2131. [[CrossRef](#)]
39. Zhou, S.; Song, W. Deep learning-based roadway crack classification with heterogeneous image data fusion. *Struct. Health Monit.* **2021**, *20*, 1274–1293. [[CrossRef](#)]
40. Loupos, K.; Doulamis, A.D.; Stentoumis, C.; Protopapadakis, E.; Makantasis, K.; Doulamis, N.D.; Amditis, A.; Chrobocinski, P.; Victores, J.; Montero, R.; et al. Autonomous robotic system for tunnel structural inspection and assessment. *Int. J. Intell. Robot. Appl.* **2018**, *2*, 43–66. [[CrossRef](#)]
41. Menendez, E.; Victores, J.G.; Montero, R.; Martínez, S.; Balaguer, C. Tunnel structural inspection and assessment using an autonomous robotic system. *Autom. Constr.* **2018**, *87*, 117–126. [[CrossRef](#)]
42. Protopapadakis, E.; Stentoumis, C.; Doulamis, N.; Doulamis, A.; Loupos, K.; Makantasis, K.; Kopsiaftis, G.; Amditis, A. Autonomous Robotic Inspection in Tunnels. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* **2016**, *III-5*, 167–174. [[CrossRef](#)]
43. Kumar, M.; Saxena, R. Algorithm and technique on various edge detection: A survey. *Signal Image Process.* **2013**, *4*, 65.
44. Lavania, K.K.; Kumar, R. Image enhancement using filtering techniques. *Int. J. Comput. Sci. Eng.* **2012**, *4*, 14.
45. Hoang, N.-D. Detection of Surface Crack in Building Structures Using Image Processing Technique with an Improved Otsu Method for Image Thresholding. *Adv. Civ. Eng.* **2018**, *2018*, 3924120. [[CrossRef](#)]
46. Gao, M.; Wang, X.; Zhu, S.; Guan, P. Detection and Segmentation of Cement Concrete Pavement Pothole Based on Image Processing Technology. *Math. Probl. Eng.* **2020**, *2020*, 1360832. [[CrossRef](#)]

47. Chen, C.; Seo, H.; Zhao, Y.; Chen, B.; Kim, J.; Choi, Y.; Bang, M. Automatic Pavement Crack Detection Based on Image Recognition. In Proceedings of the International Conference on Smart Infrastructure and Construction 2019 (ICSIC), Cambridge, UK, 8–10 July 2019; ICE Publishing: London, UK, 2019; pp. 361–369. [\[CrossRef\]](#)
48. Yang, C.; Geng, M. The crack detection algorithm of pavement image based on edge information. *AIP Conf. Proc.* **2018**, *1967*, 040023. [\[CrossRef\]](#)
49. Ahmadi, A.; Khalesi, S.; Bagheri, M. Automatic road crack detection and classification using image processing techniques, machine learning and integrated models in urban areas: A novel image binarization technique. *J. Ind. Syst. Eng.* **2018**, *11*, 85–97, No. Special issue: 14th International Industrial Engineering Conference. Available online: https://www.jise.ir/article_69387.html (accessed on 20 May 2024).
50. Arya, D.; Maeda, H.; Ghosh, S.K.; Toshniwal, D.; Sekimoto, Y. Rdd2020: An image dataset for smartphone-based road damage detection and classification. *Data Brief* **2021**, *36*, 107133. [\[CrossRef\]](#)
51. Özgenel, Ç.F.; Sorguç, A.G. Performance Comparison of Pretrained Convolutional Neural Networks on Crack Detection in Buildings. In Proceedings of the 35th International Association for Automation and Robotics in Construction, Berlin, Germany, 20–25 July 2018. [\[CrossRef\]](#)
52. Dorafshan, S.; Thomas, R.J.; Maguire, M. Comparison of deep convolutional neural networks and edge detectors for image-based crack detection in concrete. *Constr. Build. Mater.* **2018**, *186*, 1031–1045. [\[CrossRef\]](#)
53. Wang, K.; Fang, B.; Qian, J.; Yang, S.; Zhou, X.; Zhou, J. Perspective Transformation Data Augmentation for Object Detection. *IEEE Access* **2020**, *8*, 4935–4943. [\[CrossRef\]](#)
54. Chiu, M.-C.; Chen, T.-M. Applying Data Augmentation and Mask R-CNN-Based Instance Segmentation Method for Mixed-Type Wafer Maps Defect Patterns Classification. *IEEE Trans. Semicond. Manuf.* **2021**, *34*, 455–463. [\[CrossRef\]](#)
55. Shorten, C.; Khoshgoftaar, T.M. A survey on Image Data Augmentation for Deep Learning. *J. Big Data* **2019**, *6*, 60. [\[CrossRef\]](#)
56. Bhattarai, B.; Baek, S.; Bodur, R.; Kim, T.-K. Sampling Strategies for GAN Synthetic Data. In Proceedings of the ICASSP 2020–2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain, 4–8 May 2020; IEEE: New York, NY, USA, 2020; pp. 2303–2307. [\[CrossRef\]](#)
57. Frid-Adar, M.; Klang, E.; Amitai, M.; Goldberger, J.; Greenspan, H. Synthetic data augmentation using GAN for improved liver lesion classification. In Proceedings of the 2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018), Washington, DC, USA, 4–7 April 2018; IEEE: New York, NY, USA, 2018; pp. 289–293. [\[CrossRef\]](#)
58. Kukreja, V.; Kumar, D.; Kaur, A.; Geetanjali; Sakshi. GAN-based synthetic data augmentation for increased CNN performance in Vehicle Number Plate Recognition. In Proceedings of the 2020 4th International Conference on Electronics, Communication and Aerospace Technology (ICECA), Coimbatore, India, 5–7 November 2020; IEEE: New York, NY, USA, 2020; pp. 1190–1195. [\[CrossRef\]](#)

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.