automation

MDPI

Article

You Only Look Once v5 and Multi-Template Matching for Small-Crack Defect Detection on Metal Surfaces

Pallavi Dubey [1], Seth Miller [2], Elif Elçin Günay [3,4], John Jackman [5], Gül E. Kremer [3] and Paul A. Kremer [6,*]

[1] Bio Economy Institute, Iowa State University, Ames, IA 50011, USA; pallavid@iastate.edu
[2] Caterpillar Inc., Griffin, GA 30224, USA
[3] School of Engineering, University of Dayton, Dayton, OH 45409, USA; egunay1@udayton.edu (E.E.G.); gkremer2@udayton.edu (G.E.K.)
[4] Department of Industrial Engineering, Sakarya University, Sakarya 54050, Turkey
[5] Department of Industrial and Manufacturing and Systems Engineering, Iowa State University, Ames, IA 50011, USA; jkj@iastate.edu
[6] Department of Civil, Construction and Environmental Engineering, Iowa State University, Ames, IA 50011, USA
* Correspondence: pkremer@iastate.edu

Abstract: This paper compares the performance of Deep Learning (DL) and multi-template matching (MTM) models for detecting small defects. DL models extract distinguishing features of objects but require a large dataset of images. In contrast, alternative computer vision techniques like MTM need a relatively small dataset. The lack of large datasets for small metal-surface defects has inhibited the adoption of automation in small-defect detection in remanufacturing settings. This motivated this preliminary study to compare template-based approaches, like MTM, with feature-based approaches, such as DL models, for small-defect detection on an initial laboratory and remanufacturing industry dataset. This study used You Only Look Once v5 (YOLOv5) as the DL model and compared its performance against the MTM model for small-crack detection. The findings of our preliminary investigation are as follows: (i) YOLOv5 demonstrated higher performance than MTM in detecting small cracks; (ii) an extra-large variant of YOLOv5 outperformed a small-size variant; (iii) the size and object variety of the data are crucial in achieving robust pre-trained weights for use in transfer learning; and (iv) enhanced image resolution contributes to precise object detection.

Keywords: small metal-surface cracks; YOLO; multiple-template matching; bootstrapping; machine learning approaches in remanufacturing

Academic Editors: Duc Truong Pham, Zude Zhou, Quan Liu, Wenjun Xu, F. Javier Ramírez, Marcello Fera, Mario Caterino and Jeremy Rickli

Received: 1 January 2025
Revised: 3 March 2025
Accepted: 8 March 2025
Published: 7 April 2025

Citation: Dubey, P.; Miller, S.; Günay, E.E.; Jackman, J.; Kremer, G.E.; Kremer, P.A. You Only Look Once v5 and Multi-Template Matching for Small-Crack Defect Detection on Metal Surfaces. *Automation* **2025**, *6*, 16. https://doi.org/10.3390/automation6020016

1. Introduction

Remanufacturing, which brings new life to used parts, involves a thorough inspection process to ensure part restoration and avoid safety hazards [1]. The traditional method of defect detection in such processes is manual inspection by human operators. However, there are some drawbacks to manual inspection, such as it being subjective and susceptible to errors. For instance, assessments by operators may be inconsistent because of human subjectivity, fatigue, or variations in environmental conditions. Moreover, human inspectors find it challenging to detect minor surface defects on metal parts due to their subtle nature [2–4]. Small metal defects, such as cracks, may cause serious problems, depending on their location when they are not detected, such as, for example, seal loss; compression loss; or even further component, assemblies, and systems failures or engine failure. To ensure that products are restored to the as-new condition in remanufacturing, defects

in critical locations, regardless of their sizes, typically have to be repaired. Automated inspection could address the challenges of manual inspection and improve inspection efficiency in remanufacturing.

Recent developments in deep learning (DL) have provided efficient solutions for inspection processes and have opened the way toward automated defect detection. In general, DL models excel at pattern recognition and can achieve high accuracy in prediction. However, they are highly dependent on large datasets of defect images to train models effectively [5–7]. Unlike large-scale manufacturing lines, remanufacturing facilities typically handle a smaller volume and a wider variety of parts. Additionally, diverse operating conditions of the products while in service lead to variations in the defect types, locations, and sizes. Together, these factors make it difficult to acquire the significant amount of labeled data, often 200 to over 1000 images per defect category, that are needed for good-performing DL object detection models [8–11]. Furthermore, the data collection stage can be challenging in real-world remanufacturing settings due to important factors such as variations in the lighting conditions, image resolution, and image acquisition perspective [12,13].

Template Matching is an alternative approach to DL for automated defect detection, and it has been used for quality control in manufacturing [14]. Template Matching has performed well with poor contrast images and has been applied to microscopy images for object detection. To further enhance the Template Matching model performance, the model uses augmentation and Non-Max Suppression [15–20]. However, Template Matching models do not perform well when there are deviations in object shape and orientation [15]. MTM is a computer vision technique that translates different templates for patterns to be recognized across an image to identify objects that match a template. Unlike DL models, MTM does not require extensive data for effective object detection [19]; therefore, it can be considered an alternative approach to DL models when data are limited. DL models, particularly Convolutional Neural Networks (CNNs), automatically learn features from data, which enhances the robustness of computer vision systems [21–23]. They exhibit superior performance in detecting small objects. For instance, YOLOv5 has enhanced performance in identifying small targets (e.g., detecting drones that are 5% of the image size [24]). Also, YOLO models have shown higher mean average precision (mAP) than Single Shot MultiBox Detector (SSD), Faster Region-based Convolutional Neural Network (Faster R-CNN), and other feature-based object detection models for the GC10 DET metal defect dataset [11].

This study investigated the prediction performance of two computer vision algorithms, YOLOv5 and MTM, in detecting small defects under limited data availability. The algorithms are compared through the recall metric using crack defect images on metal surfaces of cast iron cylinder heads undergoing remanufacture. This study represents a first in the literature to compare a template-based approach (MTM) with a feature-based approach (YOLOv5) for small-defect detection within the context of limited remanufacturing data. Additionally, the impacts of deepening the DL model layers, increasing image resolution, and transfer learning were also investigated, as the literature suggests that these enhancements could improve model performance for large datasets [21,22].

In addition, findings from this study contribute to the literature by (i) comparing the performance of a feature-based approach and a template-based approach on a limited industry dataset; (ii) assessing the impact of network depth (a deeper convolutional neural network) on prediction performance by experimenting with two YOLOv5 variants—the small model (YOLOv5s) and the extra-large model (YOLOv5x); (iii) investigating the impact of transfer learning on the prediction capabilities of YOLOv5 models; and (iv) evaluating the impact of higher-resolution images on the prediction performance of YOLOv5. Section 2

of the paper reviews previous work on small-object detection using computer vision techniques, along with the challenges of small-object detection and limited data. Section 3 describes the methodology, Section 4 the numerical study, and Section 5 concludes the paper with a discussion of the potential directions for future work.

2. Literature Review

2.1. YOLOv5

YOLOv5 is a one-stage DL detector model that belongs to the You Only Look Once (YOLO) family of computer vision models [25,26]. The fundamental principle of all YOLO variants is to divide the image into multiple grids of cells and then predict the bounding boxes with class probabilities based on contained features. Ten variants have been released since 2016, and each variant introduced different architectural improvements to enhance prediction performance. Variants released after YOLOv5 demonstrated improved accuracy; however, they require more computational power, which can limit new variants' applications in industry deployments [27]. YOLOv5 has found wide application, ranging from traffic flow detection [28] to small-defect detection [29,30].

YOLOv5 has five different model depths: nano, small, medium, large, and extra-large. The YOLOv5s (small) model generates weights with only 14MB, 7.2 M model parameters, and 15.8 giga floating-point operations per second (GFLOPs). This leads to fast inferencing speeds for the small model, allowing quick simulation, a desired feature in automated defect detection in real-time applications. YOLOv5x, an extra-large variant, has deeper convolutional layers for feature extraction; it uses 86.7 M model parameters, along with 205.7 GFLOPs.

The YOLOv5 architecture includes three main components: backbone, neck, and head. The backbone extracts features from the images at the smallest level of detail. The neck mixes and combines these features and passes them to the head to predict the classes and the bounding boxes enclosing the objects. The backbone of the model is the Cross Stage Partial Network (CSPNet) [31]. The neck of the model is PANet [32], and it generates feature pyramids that consist of feature maps of different scales. These maps enhance the capability of the model to detect objects of various sizes.

YOLOv5 performs data augmentation using scaling, color space adjustments, and mosaic augmentation to provide a greater and richer variation of images while training the model. Mosaic augmentation is especially useful for small-object detection, as it teaches the model to recognize small objects in different localizations irrespective of any specific context [24]. Also, the auto-learning bounding box anchor feature introduced in YOLOv3 is used in YOLOv5—this feature customizes the distribution of anchor box sizes and locations based on a custom dataset rather than a preset COCO dataset [33]. Anchor boxes are preset in some DL models to a specific height and width that correspond to the objects of interest in an image dataset. The YOLOv5 model can customize the size of anchor boxes, which improves precision in localizing the object.

2.2. Template Matching and Multi-Template Matching (MTM)

Template matching is another approach for automated defect detection [34]. A cropped image of the object of concern is defined as a template, and it is then used to find a matching region within the whole image using metrics such as correlation. This involves moving across an image while calculating the difference between the pixels for the template and the image in the overlapping region one pixel at a time. In the context of template matching, correlation often refers to comparing pixel values between the template and the corresponding region in the larger image. However, the exact method of correlation can vary based on the specific algorithm or technique used for matching templates. Metrics used to measure this

difference can be the sum of squared errors, cross-correlation, normalized cross-correlation, or some other metric. The user defines a threshold percentage to determine if the object is a match or not because a perfect match may not be possible.

MTM is an extension of the template matching algorithm and uses multiple templates to be searched throughout the entire image to identify target objects based on the similarity between the template and image patches at each pixel. These templates are the representation of the target objects varied in size or position. By using these templates, multiple objects, as well as multiple categories of objects, can be detected. Overlapping detections can be eliminated through the non-maximum suppression method. The advantage of MTM over single-template matching is that it can provide additional object orientations and scales, which are important for object detection. Augmentation techniques like flipping and rotation are also performed on the training dataset to enhance the object detection capability further if the object's orientation does not match the template [19].

MTM is known to perform well for poor contrast images and uses minimal parameters for training, such as the correlation coefficient normed. Unlike methods based on detecting edges and shapes, MTM is not noise-sensitive and uses a smaller annotated dataset for the templates [19]. MTM has been successfully used in microscopy images and vegetable detection in fields where the object of interest represents a small portion of an image and is randomly located [15,16,19].

2.3. Small-Object Detection

Liu et al. [35], in a recent study, identified four primary challenges that affect the accuracy of DL models in detecting small objects. These challenges include the use of feature layers having insufficient information, poor object resolution, class imbalances, and insufficient quantities of small anchor boxes.

Feature layers having insufficient information occur when a defect is small relative to the image size because a large part of the image background does not provide meaningful information, i.e., a negative background. For example, consider the small defects outlined by yellow-green boxes in the original image at the top of Figure 1. Only a small image area includes useful information, i.e., positive foreground (yellow-green squares in Figure 1). Features are formed by the object detection model by aggregating pixels in convolutional layers (for example, the model learns the features to detect a cat). The model's predictions are based on calculated loss functions, which sum up across convolutional layers based on the difference between the prediction and the ground truth (i.e., the actual object). Because the ground truth is small for a small defect, the feature extraction will also be small when training the model. Thus, the feature layers may not contain enough information.

Image resolution can affect a model's ability to detect small defects. For instance, issues related to small-scale and weak feature responses have been addressed in other research fields, such as urban monitoring, military reconnaissance, medicine, and national security applications [22,36]. These studies highlighted the importance of high-resolution images to improve the accuracy of neural networks. Han et al. [22] concluded that higher resolution helped show more texture, shape, and additional details for a given object. However, using high-resolution images significantly increases the computation time [37].

Class imbalance can also contribute to a model's poor performance in small-defect detection. Inconsistent or incorrect annotations are major reasons for class imbalance apart from imbalanced data collection. Inconsistency or errors in annotations affect model precision, which has been the case for the COCO dataset. COCO [38] includes 328,000 images, with 41% of objects categorized as small (minimum rectangle area of bounding box less than 32×32 pixels), 34% as medium (minimum rectangle area between 32×32 and maximum 96×96 pixels), and 24% as large (anything above 96×96 pixels). Despite most data in

COCO being categorized as small, the average precision for small objects was two to three times lower than for large objects [39].

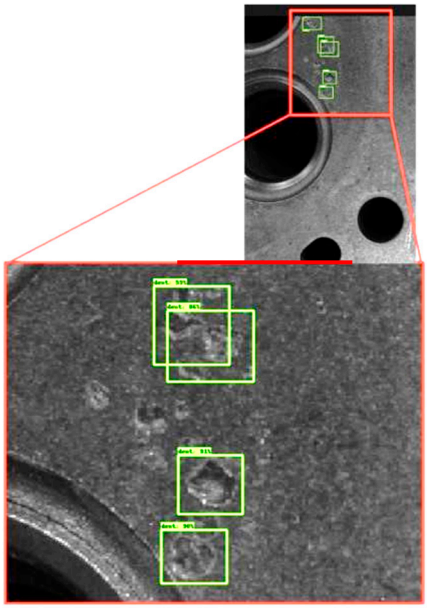


Figure 1. Examples of small defects in the original image (**top**) and an enlarged inset (**bottom**).

Anchor boxes play a significant role in small-defect detection in DL models. A DL model makes thousands of predictions in a given image and only shows the one belonging to a defined object class. It does so by creating thousands of prior boxes (anchor boxes), defined by the model creator based on the size of objects in the training dataset, their aspect ratio, or their location in different parts of the image. In the Retina Net DL configuration where the smallest anchor box size is 32 × 32, smaller objects will not be detected [40]. Ensuring the distribution and size of anchor boxes are tuned with the data under analysis is key to small-object detection. DL algorithms like YOLOv5 create custom anchor boxes based on a given training dataset [25]. This helps ensure that many sizes, shapes, and locations based on contextual information are covered.

2.4. Limited Data

The second problem addressed in this study is the impact of limited data on small-defect detection. Various techniques, such as transfer learning and data augmentation, have been used to address the limited data issue in the last decade [8,12,13]. Transfer learning, an important aspect of DL, has been found to improve DL model performance on limited data [13]. Transfer learning leverages the results of a previous model training task to enhance the outcomes of a new task, particularly by using weights from a pre-trained model to initialize a new model. The pre-trained model is typically trained with a large dataset, generating weights used for initialization for the new task [33]. For example, if a model consists of n layers, the 1st layer learns the easiest pattern/feature in the image, and as we progress to the nth layer, the information/feature learned becomes more complex. This information is saved as weights. Weight initialization is performed before training the model. Good initial weights can improve the convergence of a neural network. Convergence refers to the point at which the training process reaches a stable state, and the weights and biases have settled on values that output accurate predictions for training data. A pre-trained model using a large dataset (COCO in our case) can be useful for a broad range of object detection contexts. Also, using pre-trained model weights saves time. Training a model without pre-trained weights can take several days or weeks to fully train [41].

Data augmentation is another technique that can improve the accuracy of models trained on limited data [16,19,22]. The augmentation process generates new training data by applying transformations to the original input data. This helps generalize the training data and reduces the difference in distribution between training and test sets. Several augmentation techniques have been used in the literature; e.g., the Gaussian mixture model, Generative Adversarial Networks, and classical augmentation techniques such as rotation, flip, blur, mosaic, and color space adjustments [11,22,23,42]. Previous work suggests transfer learning and data augmentation help improve precision and recall for small-object detection with limited data [43,44].

This study examined multiple factors and methods to determine to what extent limited data can be used to achieve an acceptable defect detection accuracy (95% or above) for small defects to aid the quality inspection process and minimize rework in a remanufacturing assembly line. The study first explored the use of a DL model where transfer learning is used to address the limited data issue. A simpler method based on MTM, which requires fewer images, was also investigated. Results from both methods were compared to determine their relative performance on the same images acquired in conditions of poor illumination and with variable camera perspective settings with no attempt to correct for distortion in the images stemming from the cameras used to acquire the images. The impact of image resolution and model depth on the performance of a DL model trained on a limited dataset was also examined. In the next section, the methodology used in this study is discussed.

3. Methodology

The overall flow of the methodology is presented in Figure 2. The initial step is data preparation, which involves the pre-processing of an industry-specific dataset and the open-source GC10 DET dataset for training [11]. GC10 DET was included specifically to investigate whether pre-trained weights obtained through transfer learning improve YOLOv5 model prediction performance; therefore, it was only used for transfer learning purposes. To estimate YOLOv5 models' prediction performance, a bootstrapping approach is incorporated to handle limited and imbalanced data issues. Next, numerical studies were conducted to investigate the impact of model complexity, transfer learning, and image resolution on YOLOv5's prediction performance, as well as to compare the performance of YOLOv5 with MTM. Lastly, the effectiveness of each model was evaluated. The details of each step in Figure 2 are explained in Sections 3.1–3.4.

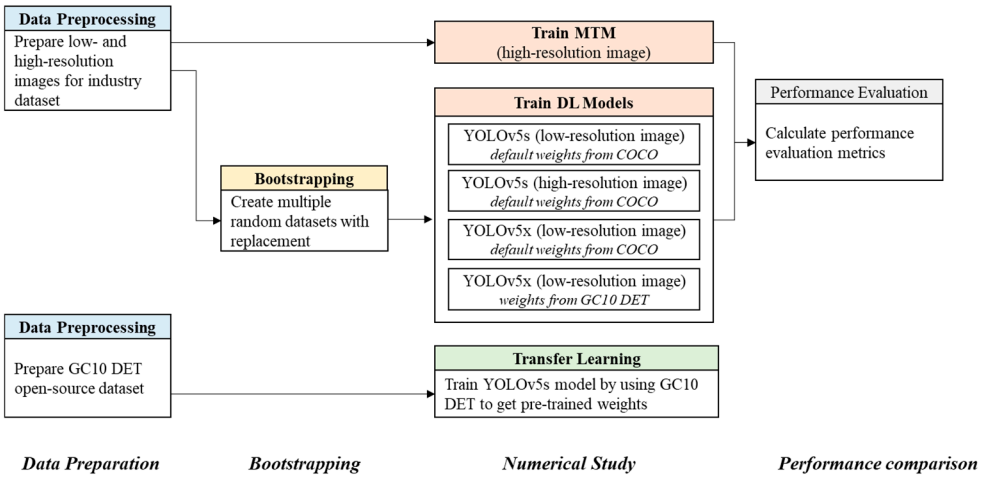


Figure 2. The overall flow of the methodology.

3.1. Dataset Preparation

The specifications of the two different datasets: (i) industry data and (ii) GC10 DET open-source dataset, are detailed in Sections 3.1.1 and 3.1.2, respectively.

3.1.1. Industry Dataset

Our investigation focused on developing a system to detect small defects, specifically, cracks no larger than ~2 mm width and ~6 mm in length, on the surfaces of used cylinder heads. To capture these defects, we employed two distinct image acquisition methods and settings: (i) laboratory setup and (ii) in process at the remanufacturer's facility. The laboratory method ensures more controlled image acquisition, while the in process method is less controlled but is required to increase the number of images containing small defects for model training.

(i) Laboratory setup: A Basler ace 3088-16 gm, monochrome area scan camera with an Edmund Optics 8 mm HP series lens was used along with custom C++ code to automate the image acquisition. The camera was positioned at a 100 mm working distance from the cylinder head, with a fixed f/4 aperture, 8 bit pixel depth, and perpendicular alignment for all images, using a Techman TM5-900 cobot. The camera system captured images at a resolution of 3088 × 2064 pixels, corresponding to approximately 28 pixels/mm. Illumination relied on ambient lighting conditions (fluorescent lighting), and the camera employed no filters to mitigate spectral interference from the ambient lighting. Images were captured at the center of each pre-defined grid box measuring 110.28 mm × 74.33 mm on the cylinder head surfaces. No camera models to correct for distortions created by the camera perspective and lens elements were applied to these images. The cast iron cylinder heads were sampled and provided by a remanufacturer in the "cleaned" state produced in the remanufacturing process to enable the manual methods used for defect inspection.

(ii) Remanufacturer's facility: A standard SLR camera was used with a resolution of 2592 × 1944 pixels and was mounted on a manual slide system to control the camera's position relative to the cylinder head surface. As with the laboratory setup, cylinder heads were in the "cleaned" state and the camera relied on ambient fluorescent lighting for illumination.

For this investigation, images were captured using both methods and settings in ambient fluorescent lighting, and were merged into a single dataset. This resulted in a total of 103 images containing cracks. To facilitate defect detection model development, three inspectors independently annotated the location and type of defects within the images. The open-source labelImg (https://github.com/heartexlabs/labelImg, accessed on 10 December 2022) software aided the annotation process. To minimize errors and bias, each inspector reviewed the same set of images consecutively. They followed an annotation protocol that prioritized:

- Tight Bounding Boxes—minimize noise by drawing bounding boxes as tightly as possible around the defects;
- Complete Labeling—ensure all defects are labeled individually, avoiding any grouping;
- No Missed Defects—meant to leave no defect left unlabeled.

The labeling process continued until consensus was reached by all inspectors employing this protocol [45].

Resizing images is a critical pre-processing step in computer vision applications for reducing the computing time needed for training the algorithms. The original images were resized to 640 × 640 pixels for YOLOv5s and YOLOv5x. Prior research has shown improvement in small-object detection with high-resolution images [22]. Therefore, a

separate high-resolution dataset was created by partitioning the original image into blocks of 640 × 640 pixel sub-images, as shown with red boxes in Figure 3, while keeping the original resolution intact, i.e., 3088 × 2064. For partitioning the images, an OpenCV library was used [46]. The crack in the magnified region of the image at left in Figure 3 is in the yellow bounding box.

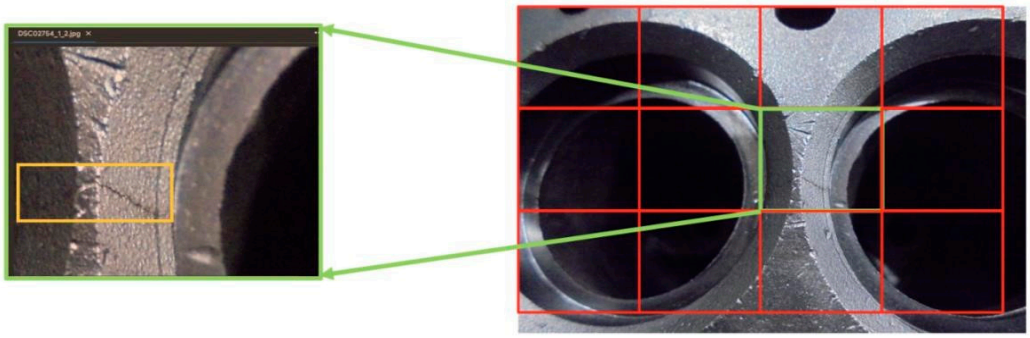


Figure 3. Example showing the 640 × 640 pixel sub-images derived from a large-size, 3088 × 2064 pixel image and an example crack defect identified in one of the partitions.

3.1.2. Benchmark Datasets for Transfer Learning

The pre-trained weights of YOLOv5s and YOLOv5x, trained by Jocher et al. [25] on the COCO dataset, were used as initial weights for training models on our custom dataset. The other benchmark dataset GC10 DET [11], needed for transfer learning, has ten categories of metal-surface defects and 2272 images. The reason for choosing the GC10 DET dataset is that it possesses a diverse range of images, including various metal defects of varied type, size, and shape. The rationale for using GC10 DET for transfer learning was to allow the model to learn the features of defects similar to those in our industry dataset. By using these pre-trained weights, we aimed to streamline the training process and enhance the prediction performance of the model, YOLOv5.

There were three important issues we found related to defect labels in the GC10 DET dataset: (i) some bounding boxes tightly captured the defects, while some were loosely drawn, leading to noise; (ii) some images were missing defect annotations; and (iii) some images in the dataset exhibited inconsistencies in the labeling of the small defects, i.e., some were labeled as a group, whereas the others were individually labeled. The GC10 DET data underwent a pre-processing stage to correct labels on images and revise bounding boxes on images to have the same labeling strategy as the custom dataset [45] mentioned in Section 3.1.1.

3.2. Bootstrapping

The bootstrap method was used to provide a reliable estimate of the performance metrics, because the use of the best-performing result can bias the model's performance toward an optimistic estimate given the small dataset [47–49]. Small datasets are susceptible to outliers and noise. Additionally, limited data often result in high variation in prediction results, making model performance less reliable [13]. In this study, bootstrapping was used to address the small dataset issue and accurately capture the prediction performance of YOLOv5 models.

In this method, a bootstrap sample is randomly obtained from a pool of a given population with replacement, ensuring that a sample (image in our case) has an equal chance of being picked in subsequent draws from the pool. Through replicating the sampling iteration, multiple bootstrap samples are created, which are the same size as the original dataset, i.e., 103 images for our problem. This approach, having multiple resampling iterations, ensures that each bootstrap sample has a different summary statistic

for the performance metric, which is important for calculating the confidence interval [50]. The samples not selected in the training dataset (~30%) were then divided in half for validation and testing datasets. Each dataset was used to estimate prediction performance metrics. The number of iterations (i.e., bootstrap samples) typically varies between 200 and 1000 [50,51]. Our study started with 50 iterations and then increased until the width of the confidence interval was on the order of 1% at 500 samples. Increasing the number of iterations provides a better estimate of the sampling distribution. Two metrics were calculated using the 500 bootstrap samples: average precision and recall for each dataset. These two metrics are the estimators used to predict the average precision and recall values, respectively. The twelve largest and the twelve smallest estimates for each metric were removed before calculating the 95% confidence interval using the *t*-statistic [50].

3.3. Numerical Study

In the numerical study, five experiments were conducted to understand the effects of: (i) model complexity, (ii) transfer learning, and (iii) image resolution on YOLOv5 prediction performance; and (iv) to compare the performance of YOLOv5 versus MTM. Table 1 summarizes image size, algorithms used, and the dataset used for transfer learning in these experiments. In Table 1, the first two experiments, #1 and #2, investigated the effect of increasing network depth by extending the convolutional layer structure from YOLOv5s to YOLOv5x. To explore the advantage of transfer learning on prediction performance, Experiment #3 was conducted. Experiment #4 was designed to investigate the impact of input image resolution on YOLOv5s prediction performance. The last experiment, Experiment #5, was conducted to observe the prediction performance of the MTM model through comparison against that of YOLOv5 models. In all five experiments, the hyperparameters were set to a batch size of 16, a learning rate of 0.003, and training for 2499 epochs. These hyperparameter values were selected based on their superior performance after testing various preliminary training iterations.

Table 1. Summary of numerical experiments.

Experimental Structure			
Experiment Number	**Image Dataset Used by the Model**	**Algorithm Used**	**Dataset Used for Transfer Learning**
1	Resized 640 × 640	YOLOv5s	COCO
2	Resized 640 × 640	YOLOv5x	COCO
3	Resized 640 × 640	YOLOv5s	GC10
4	High-resolution (3088 × 2064)	YOLOv5s	COCO
5	High-resolution (3088 × 2064)	MTM	N/A

3.3.1. Model Size

Experiments #1 and #2 investigated the impact of model complexity on the model's prediction performance using two variants of YOLOv5: YOLOv5s and YOLOv5x. Both experiments used default pre-trained weights derived from the COCO dataset. In both experiments, the model converged at about ~250 epochs. The YOLOv5s and YOLOv5x were trained on a NOVA high-performance computing cluster, and it took less than an hour to train each model. For a more detailed explanation of the architecture and loss functions used, please refer to Jocher et al. [25].

3.3.2. Transfer Learning

Experiment #3 in Table 1 was conducted to investigate the impact of transfer learning. In this context, the GC10 DET dataset was used as an alternative to the COCO dataset to derive pre-trained weights, which were subsequently used during the YOLOv5 models training phase for the industry dataset. Resized GC10 DET images (640 × 640 pixels) were used to create 500 bootstrap samples for the training phase.

3.3.3. Image Resolution

Experiment #4 was conducted to assess the impact of higher-resolution images on DL algorithm performance. In this experiment, YOLOv5s was trained on the high-resolution image dataset, using default pre-trained weights from the COCO dataset.

3.3.4. MTM

In Experiment #5, MTM was used to identify defects in the higher-resolution images. MTM defect detection was calculated using the Coefficient Correlation Normed $R(x,y)$ to make predictions about the test image, which is determined by

$$R(x,y) = \frac{\sum_{x',y'} T'(x',y')\ I'(x+x',\ y+y')}{\sqrt{\sum_{x',y'} \cdot T'(x',y')^2 \cdot \sum_{x',y'}\ I'(x+x',\ y+y')^2}} \quad (1)$$

where T' represents the template's mean-subtracted coordinates, and I' represents the coordinates on the test image used to locate the template with reference to the top left corner of the image. The numerator computes the cross-correlation between the mean-subtracted template T' and the mean-subtracted image patch I'. The denominator normalizes the cross-correlation by the product of the squared norms energies of T′ and I′, where x and y represent the original data values, and x' and y' represent the mean-subtracted version of the original data. Four other metrics are available for evaluating MTM performance, but the Coefficient Correlation Normed metric was used because it yielded the best results for our dataset [52]. The model was trained on all defect templates, excluding one as a test sample (i.e., training set of 102 images and test set with 1 image) to check whether it could identify the defect in the given test image. A total of 103 iterations were performed by alternatively evaluating detection on all images. The result of each iteration is a Bernoulli random variable that can take two possible values, either detected or undetected. Detection is when the R score is highest for a given prediction. Recall was estimated by dividing the number of true positives detected out of all iterations by 103 (true positive + false negative), where the true positive is the highest R score amongst the detected objects, which can range from 0.5 to 1 andfalse negative is when the model is unable to detect a crack. For MTM, there were two hyperparameters; namely, a recommended score threshold of 0.5 [19] and the number of predictions (N) per image defined as two because few images have more than one defect. This parameter tells the model to output the top two bounding box predictions with the highest score values based on Equation 1. Bootstrapping was not used to calculate the confidence interval because bootstrapping would reduce the images in the training set. This reduction would lower the model's performance because fewer templates (less than 102) would be available to map to the given test image.

3.4. Performance Comparison

To evaluate the performance of an object detection algorithm, key metrics like precision, recall, and average precision are commonly used. Before defining these metrics, it is essential to explain Intersection over Union (IoU) and confidence score, which are critical to calculating key metrics. IoU is the ratio of the ground truth bounding box for a defect

that is contained in the predicted bounding box generated by a DL algorithm to the union of predicted and ground truth. The ground truth bounding boxes for all images are pre-determined by a subject matter expert before evaluating the algorithms' prediction performance. The confidence score shows the probability of the image being detected correctly by an algorithm.

Predictions are considered true positives when both IoU and the confidence scores are above threshold values; otherwise, they are false positives. A false positive represents a Type I error, while a false negative corresponds to a Type II error. In this study, a fixed threshold value of 0.5 was used for IoU. The confidence threshold is set by determining the confidence score where the F1-score (harmonic mean of precision and recall) is a maximum, which is calculated by the YOLOv5 algorithm.

Recall is the ratio of the number of true positives to the total number of actual objects in the image. Recall is used to evaluate how well a model performs in predicting all the actual objects. Precision is the proportion of true positives to the total number of positive predictions. If the model predicts false positives, the precision will decrease. Average precision (AP) is the area under the precision-recall curve based on the defined confidence threshold and the intersection over union threshold for a single object category.

4. Results

Table 2 summarizes the AP and recall values of experiments #1 to #5. Sections 4.1–4.4 explore the impact of model size, transfer learning, and image resolution and compare the performance of the YOLOv5 and MTM models, respectively.

Table 2. Result summary for all experiments.

Experiment Number	Dataset Used by the Model	Algorithm Used	Data Used for Transfer Learning	95% Confidence Intervals for AP	95% Confidence Intervals for Recall
1	Resize 640 × 640	YOLOv5s	COCO	93.90–94.83%	90.50–91.76%
2	Resize 640 × 640	YOLOv5x	COCO	94.08–95.22%	94.70–95.75%
3	Resize 640 × 640	YOLOv5s	GC10	85.16–87.01%	84.66–86.52%
4	High-resolution	YOLOv5s	COCO	95.14–95.89%	92.82–93.97%
5	High-resolution	MTM	N/A	N/A	12.37%

4.1. Impact of Model Size

We compared two YOLO variants, YOLOv5s and YOLOv5x, to assess how increasing the depth of the model's network affects its prediction performance in Experiments #1 and #2 in Table 2. Both models, equipped with COCO pre-trained weights, performed well for small-defect detection on our limited dataset. Although there was no significant difference for AP, YOLOv5x showed a significant improvement in recall. The relationship between network depth and performance metrics like recall and AP can be influenced by various factors. Generally, deeper CNNs tend to improve recall because they capture intricate details, which enhances recognition of a broader range of instances and positively impacts recall [21]. However, the improvement in recall does not necessarily translate to more precise localization, which is necessary to increase AP. Further, the recall metric emphasizes retrieving relevant instances regardless of precision, which might show improvements due to the broader generalization by deeper layers [40]. Thus, our preliminary results show that training a deeper CNN yields better performance in reducing false negatives and a higher chance of detecting the maximum number of defects present.

4.2. Impact of Transfer Learning

A comparative analysis of Experiments #1 and #3 in Table 2 shows that the transfer learning performed with the GC10 DET dataset resulted in poorer performance, with recall and AP values dropping 6–10%. This decrease could be associated with the size of the datasets. Given that the ~328,000 images in the COCO dataset is much larger than the ~2272 images in GC10 DET, the initial weights are calculated by observing a wide range of instances and capturing various instances, which are more robust and generalized by first employing the COCO dataset. Even though GC10 DET shares similarities with the industry data, its small size might not capture sufficient variety, which may produce ineffective initial weights that do not perform well across different instances. This suggests the importance of having sufficient data for the initial weights. Nonetheless, the amount of data for sufficiency was not assessed in this study.

4.3. Impact of Resolution

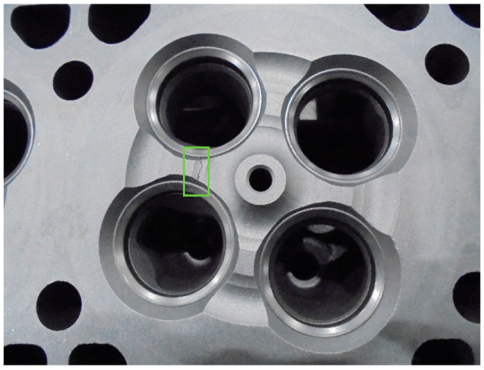
The analysis of Experiments #1 and #4 demonstrates the impact of image resolution on the model's prediction performance. The results show modest differences in AP and recall values, indicating that YOLOv5s with high-resolution images outperforms YOLOv5s with resized images. Further study is needed to explore the trade-offs between image resolution and model performance for rapid identification of small defects.

4.4. Performance Comparison Between MTM and YOLOv5

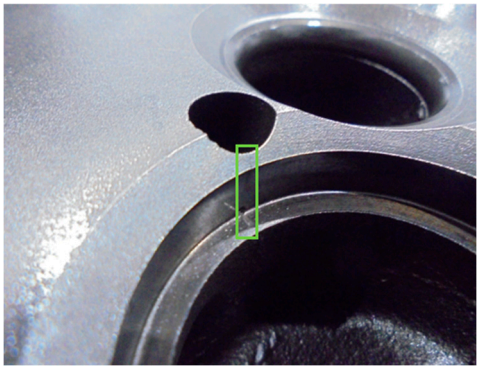
The results from Experiment #5 show that MTM performed poorly at crack detection, detecting only 12.37% of true positives out of the total 103 crack defects in the industry dataset. The precision metric is not calculated in MTM because it provides only Bernoulli outputs; i.e., detection (1) and no detection (0). Hence, we could not calculate AP because it does not account for false positives.

Figure 4 presents example prediction results from YOLOv5s and MTM models and compares them with the ground truth in the images shown. Due to the small defect sizes, the prediction results for MTM and YOLOv5 are presented in cropped and enlarged images in Figure 4 to improve the visibility of defects in the figure. MTM was able to detect the defects that are relatively larger and easy to detect; however, it was not able to identify the smaller defects (see MTM prediction for image #2 in Figure 4).

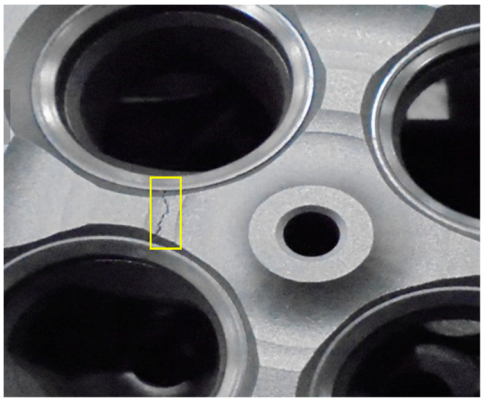
Figure 5 presents the prediction results of YOLOv5 with varying illumination, camera perspective, and image focus/depth of field. In contrast to MTM, YOLOv5 was robust enough in extracting the features and detecting defects across varying conditions. Lower confidence scores were observed for the larger defects, which were underrepresented in the training data set. Because the dataset was comprehensive enough to incorporate various small-size defects, YOLOv5 correctly identified the small defects with high confidence scores.



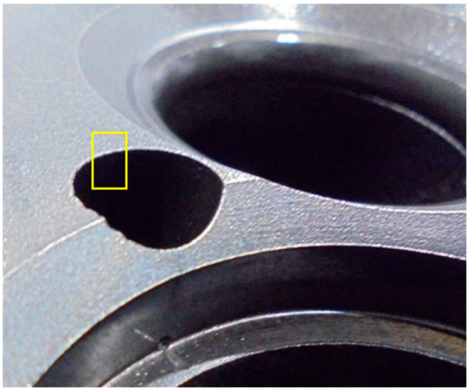
Ground truth for image #1



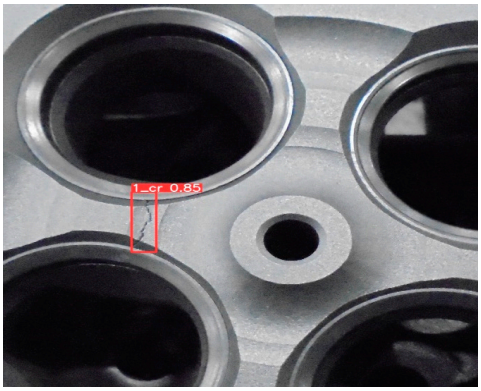
Ground truth for image #2



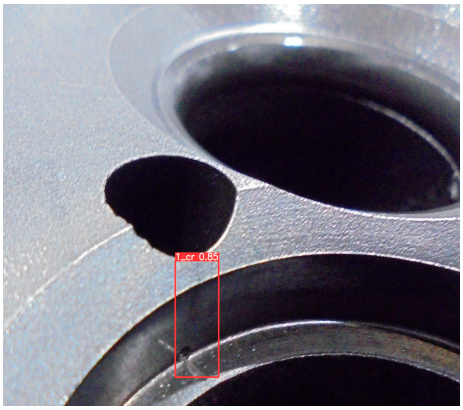
MTM for image #1 (cropped and enlarged)



MTM for image #2 (cropped and enlarged)

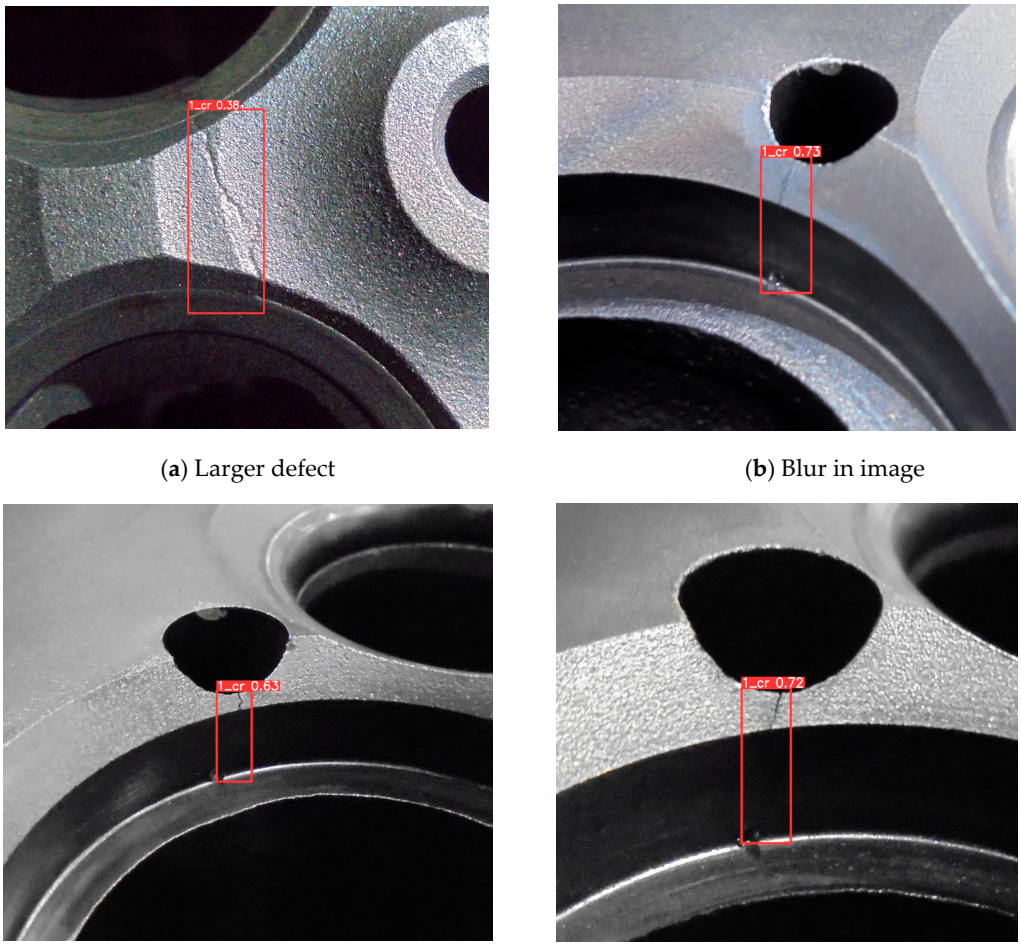


YOLOv5 for image #1 (cropped and enlarged)



YOLOv5 for image #2 (cropped and enlarged)

Figure 4. Sample images with ground truth, MTM prediction (Experiment #5), and YOLOv5 prediction (Experiment #1).



(**a**) Larger defect (**b**) Blur in image

(**c**) Small defects under low illumination

Figure 5. Sample prediction results under varied imaging conditions for YOLOv5.

5. Conclusions and Future Work Direction

In this study, we evaluated the performance of the DL object detection model YOLOv5 and the template matching model MTM using industry data to detect cracks on metal surfaces. YOLOv5x had the best performance for crack detection, with a 95% confidence interval for recall of 94.70–95.75%, compared to a 12.37% average recall for MTM. Varying illumination conditions, camera perspective, defect scales, and object distortion added complexity to the dataset. These represent realistic conditions found when relatively low-cost image acquisition means are employed in remanufacturing settings. When consideration is given to these conditions, the small objects of interest (i.e., cracks in this investigation) and the limited data (i.e., defects available for model training), defect detection becomes more challenging. Our preliminary findings clearly indicate that YOLOv5s and YOLOv5x are capable of crack detection. Higher-resolution input data exhibited some small improvements in the AP and recall performance of the YOLOv5s algorithm. Also, deepening the layers of the model yielded a better recall result but did not have a statistically significant impact on AP.

Our results show that: (i) YOLOv5 models are better suited for metal-surface crack detection than MTM; (ii) YOLOv5s and YOLOv5x were able to detect small cracks with limited data and achieved AP values of around 95%, with YOLOv5x achieving higher recall; (iii) YOLOv5x, a deeper convolutional neural network, enhanced defect detection by decreasing false positives and increasing recall; (iv) the size and variety of the objects within the dataset (GC10 DET vs. COCO) impact the robustness of the pre-trained weights,

which are used in transfer learning; and (v) higher-resolution images increased model prediction capabilities, leading to small increases in both AP and recall metrics.

A notable practical implication of our findings is the potential for deploying YOLOv5 for small-crack detection on metal surfaces. Considering its fast detection times and precise prediction, it can increase the efficiency of the inspection stage in remanufacturing and be deployed readily for a real-time cloud- or edge-based hardware implementation on the remanufacturing floor. However, extending the dataset to cover a wide range of crack types, and controlling the illumination, image acquisition perspective, and resolution, along with a few other factors, are critical to providing more generalizable insights about the impacts of these factors on prediction performance. The implications of more controlled image acquisition, along with the relative capabilities of varied YOLO model implementations for defect detection, are aspects of an ongoing investigation by the authors.

Additionally, experimenting with different defect types to explore how YOLOv5 performance changes when multiple defects are involved is important for future studies. Another avenue for future research is the investigation of the effectiveness of active learning to support the data pre-processing stage. In active learning, a learning algorithm (e.g., YOLOv5) is first trained on a smaller dataset. Then, the model queries the user to label data categories where the algorithm performs poorly by proactively selecting samples to be labeled from the pool of unlabeled data. This type of learning helps improve prediction accuracy while using a small number of training datasets, allowing the algorithm to select the most informative data to include in the dataset. In summary, the practical implementation of YOLOv5 and other YOLO variants presents a significant advancement for industrial defect detection and inspection in remanufacturing. Moreover, the ongoing iterative work of the authors evaluating continuous data collection and processing strategies involving active learning, varied data augmentation schemes, Monte Carlo dropout, and other techniques will evaluate whether these techniques can help to overcome issues with small datasets to aid in model training, minimizing bounding box uncertainty, and improving overall model effectiveness demonstrably.

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

Funding: This material is based upon work supported by the U.S. Department of Energy's Office of Energy Efficiency and Renewable Energy (EERE) under the Advanced Manufacturing Office Award Number DE-EE0007897–RM05 awarded to the REMADE Institute, a division of Sustainable Manufacturing Innovation Alliance Corp. The computing support for the research reported in this paper was partly supported by two National Science Foundation grants, MRI1726447 and MRI2018594. The Philip and Virginia Sproul Professorship at Iowa State University also partly funded the research.

Data Availability Statement: The datasets presented in this article are not readily available because they are part of multiple datasets being curated as part of an ongoing study. Requests to access the datasets should be directed to the corresponding author.

Acknowledgments: We would like to thank John Deere Reman in Springfield, MO for their support.

Conflicts of Interest: Author Seth Miller is affiliated to Caterpillar. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AP	Average precision
COCO	Common Object in Context dataset
CNNs	Convolutional Neural Networks
CSPNet	Cross Stage Partial Network
DL	Deep Learning
FPN	Feature Pyramid Network
GC10 DET	Metallic Surface Defect Detection dataset
GFLOPs	Giga Floating-point Operations per second
MTM	Multi-Template Matching
PANet	Path Aggregation Network
SSD	Single Shot MultiBox Detector
YOLOv5	You Only Look Once version 5

References

1. Kujawińska, A.; Vogt, K. Human factors in visual quality control. *Manag. Prod. Eng. Rev.* **2015**, *6*, 25–31. [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. Cha, Y.; Choi, W.; Büyüköztürk, O. Deep learning-based crack damage detection using convolutional neural networks. *Comput.-Aided Civ. Infrastruct. Eng.* **2017**, *32*, 361–378. [CrossRef]
4. Nwankpa, C.; Eze, S.; Ijomah, W.; Gachagan, A.; Marshall, S. Achieving remanufacturing inspection using deep learning. *J. Remanufacturing* **2021**, *11*, 89–105. [CrossRef]
5. Zheng, H.; Kong, L.X.; Nahavandi, S. Automatic inspection of metallic surface defects using genetic algorithms. *J. Mater. Process. Technol.* **2002**, *125–126*, 427–433. [CrossRef]
6. Kim, C.-W.; Koivo, A.J. Hierarchical classification of surface defects on dusty wood boards. *Pattern Recognit. Lett.* **1994**, *15*, 713–721. [CrossRef]
7. Kang, G.-W.; Liu, H.-B. Surface defects inspection of cold rolled strips based on neural network. In Proceedings of the 2005 International Conference on Machine Learning and Cybernetics, Guangzhou, China, 18–21 August 2005; IEEE: New York, NY, USA, 2005; Volume 8, pp. 5034–5037. [CrossRef]
8. Ferguson, M.; Ak, R.; Lee, Y.-T.T.; Law, K.H. Detection and segmentation of manufacturing defects with convolutional neural networks and transfer learning. *Smart Sustain. Manuf. Syst.* **2018**, *2*, 137–164. [CrossRef]
9. Gai, X.; Ye, P.; Wang, J.; Wang, B. Research on defect detection method for steel metal surface based on deep learning. In Proceedings of the 2020 IEEE 5th Information Technology and Mechatronics Engineering Conference (ITOEC), Chongqing, China, 12–14 June 2020; IEEE: New York, NY, USA, 2020; pp. 637–641. [CrossRef]
10. Cheng, L.; Gong, P.; Qiu, G.; Wang, J.; Liu, Z. Small defect detection in industrial x-ray using convolutional neural network. In *Pattern Recognition and Computer Vision*; Springer: Cham, Switzerland, 2019; pp. 366–377. [CrossRef]
11. Lv, X.; Duan, F.; Jiang, J.; Fu, X.; Gan, L. Deep metallic surface defect detection: The new benchmark and detection network. *Sensors* **2020**, *20*, 1562. [CrossRef]
12. Zhiyi, H.; Haidong, S.; Lin, J.; Junsheng, C.; Yu, Y. Transfer fault diagnosis of bearing installed in different machines using enhanced deep auto-encoder. *Measurement* **2020**, *152*, 107393. [CrossRef]
13. Han, T.; Liu, C.; Wu, R.; Jiang, D. Deep transfer learning with limited data for machinery fault diagnosis. *Appl. Soft Comput.* **2021**, *103*, 107150. [CrossRef]
14. Aksoy, M.S.; Torkul, O.; Cedimoglu, I.H. An industrial visual inspection system that uses inductive learning. *J. Intell. Manuf.* **2004**, *15*, 569–574. [CrossRef]
15. Tang, F.; Tao, H. Fast multi-scale template matching using binary features. In Proceedings of the 2007 IEEE Workshop on Applications of Computer Vision (WACV '07), Austin, TX, USA, 21–22 February 2007; IEEE: New York, NY, USA, 2007; p. 36. [CrossRef]
16. Bao, G.; Cai, S.; Qi, L.; Xun, Y.; Zhang, L.; Yang, Q. Multi-template matching algorithm for cucumber recognition in natural environment. *Comput. Electron. Agric.* **2016**, *127*, 754–762. [CrossRef]
17. Mahalakshmi, T.; Muthaiah, R.; Swaminathan, P. An overview of template matching technique in image processing. *Res. J. Appl. Sci. Eng. Technol.* **2012**, *4*, 5469–5473.

18. Kong, Q.; Wu, Z.; Song, Y. Online detection of external thread surface defects based on an improved template matching algorithm. *Measurement* **2022**, *195*, 111087. [CrossRef]
19. Thomas, L.S.V.; Gehrig, J. Multi-template matching: A versatile tool for object-localization in microscopy images. *BMC Bioinform.* **2020**, *21*, 44. [CrossRef]
20. Zeng, N.; Wu, P.; Wang, Z.; Li, H.; Liu, W.; Liu, X. A small-sized object detection oriented multi-scale feature fusion approach with application to defect detection. *IEEE Trans. Instrum. Meas.* **2022**, *71*, 3507014. [CrossRef]
21. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A.; Liu, W.; et al. Going deeper with convolutions. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; IEEE: New York, NY, USA, 2015; pp. 1–9. [CrossRef]
22. Han, W.; Chen, J.; Wang, L.; Feng, R.; Li, F.; Wu, L.; Tian, T.; Yan, J. Methods for small, weak object detection in optical high-resolution remote sensing images: A survey of advances and challenges. *IEEE Geosci. Remote Sens. Mag.* **2021**, *9*, 8–34. [CrossRef]
23. Chen, S.; Tang, Y.; Zou, X.; Huo, H.; Hu, K.; Hu, B.; Pan, Y. Identification and detection of biological information on tiny biological targets based on subtle differences. *Machines* **2022**, *10*, 996. [CrossRef]
24. Dadboud, F.; Patel, V.; Mehta, V.; Bolic, M.; Mantegh, I. Single-stage UAV detection and classification with YOLOV5: Mosaic data augmentation and PANet. In Proceedings of the 2021 17th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), Washington, DC, USA, 16–19 November 2021; IEEE: New York, NY, USA, 2021; pp. 1–8. [CrossRef]
25. Jocher, G.; Chaurasia, A.; Qiu, J. YOLO by Ultralytics. Version 8.0.0. Available online: https://github.com/ultralytics/ultralytics (accessed on 16 January 2024).
26. Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
27. Zhao, R.; Wang, K.; Xiao, Y.; Gao, F.; Gao, Z. Leveraging Monte Carlo dropout for uncertainty quantification in real-time object detection of autonomous vehicles. *IEEE Access* **2024**, *12*, 33384–33399. [CrossRef]
28. Sun, F.; Li, Z.; Li, Z. A traffic flow detection system based on YOLOv5. In Proceedings of the 2021 2nd International Seminar on Artificial Intelligence, Networking and Information Technology (AINIT), Shanghai, China, 15–17 October 2021; IEEE: New York, NY, USA, 2021; pp. 458–464. [CrossRef]
29. Li, J.; Su, Z.; Geng, J.; Yin, Y. Real-time detection of steel strip surface defects based on improved YOLO detection network. *IFAC-Pap.* **2018**, *51*, 76–81. [CrossRef]
30. Zeqiang, S.; Bingcai, C. Improved Yolov5 algorithm for surface defect detection of strip steel. In *Artificial Intelligence in China*; Springer: Singapore, 2022; pp. 448–456. [CrossRef]
31. Wang, C.-Y.; Liao, H.-Y.M.; Wu, Y.-H.; Chen, P.-Y.; Hsieh, J.-W.; Yeh, I.-H. CSPNet: A new backbone that can enhance learning capability of CNN. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, Seattle, WA, USA, 13–19 June 2020.
32. Liu, S.; Qi, L.; Qin, H.; Shi, J.; Jia, J. Path aggregation network for instance segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018; pp. 8759–8768.
33. Damacharla, P.; Rao, A.; Ringenberg, J.; Javaid, A.Y. TLU-Net: A deep learning approach for automatic steel surface defect detection. In Proceedings of the 2021 International Conference on Applied Artificial Intelligence (ICAPAI), Halden, Norway, 19–21 May 2021; IEEE: New York, NY, USA, 2021; pp. 1–6. [CrossRef]
34. Talmi, I.; Mechrez, R.; Zelnik-Manor, L. Template matching with deformable diversity similarity. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 175–183.
35. Liu, Y.; Sun, P.; Wergeles, N.; Shang, Y. A survey and performance evaluation of deep learning methods for small object detection. *Expert Syst. Appl.* **2021**, *172*, 114602. [CrossRef]
36. Bochkovskiy, A.; Wang, C.-Y.; Liao, H.-Y.M. YOLOv4: Optimal speed and accuracy of object detection. *arXiv* **2020**, arXiv:2004.10934.
37. McKeown, D.M.; Denlinger, J.L. Cooperative methods for road tracking in aerial imagery. In Proceedings of the Proceedings CVPR '88: The Computer Society Conference on Computer Vision and Pattern Recognition, Ann Arbor, MI, USA, 5–9 June 1988; IEEE: New York, NY, USA, 1988; pp. 662–672. [CrossRef]
38. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft COCO: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, 6–12 September 2014, Proceedings, Part v 13*; Springer International Publishing: Cham, Switzerland, 2014; pp. 740–755. [CrossRef]
39. Kisantal, M.; Wojna, Z.; Murawski, J.; Naruniec, J. Augmentation for small object detection. *arXiv* **2019**, arXiv:1902.07296.
40. Lin, T.-Y.; Goyal, P.; Girshick, R.; He, K.; Dollar, P. Focal loss for dense object detection. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 2980–2988.
41. Yosinski, J.; Clune, J.; Bengio, Y.; Lipson, H. How transferable are features in deep neural networks? In Proceedings of the 28th Conference on Neural Information Processing Systems (NIPS), Montreal, QC, Canada, 8–13 December 2014; pp. 1–9.

42. Hu, H.; Gu, J.; Zhang, Z.; Dai, J.; Wei, Y. Relation networks for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018; pp. 3588–3597.
43. Liu, J.; Guo, F.; Gao, H.; Li, M.; Zhang, Y.; Zhou, H. Defect detection of injection molding products on small datasets using transfer learning. *J. Manuf. Process.* **2021**, *70*, 400–413. [CrossRef]
44. Gong, Y.; Luo, J.; Shao, H.; He, K.; Zeng, W. Automatic defect detection for small metal cylindrical shell using transfer learning and logistic regression. *J. Nondestr. Eval.* **2020**, *39*, 24. [CrossRef]
45. Dubey, P.; Miller, S.; Günay, E.E.; Jackman, J.; Kremer, G.E.; Kremer, P.A. Deep learning-powered visual inspection for metal surfaces—Impact of annotations on algorithms based on defect characteristics. *Adv. Eng. Inform.* **2024**, *62*, 102727. [CrossRef]
46. Bradski, B. The OpenCV library. *Dr. Dobb's J. Softw. Tools Prof. Program.* **2000**, *25*, 120–123.
47. Liu, R.; Huang, M.; Gao, Z.; Cao, Z.; Cao, P. MSC-DNet: An efficient detector with multi-scale context for defect detection on strip steel surface. *Measurement* **2023**, *209*, 112467. [CrossRef]
48. Tian, R.; Jia, M. DCC-CenterNet: A rapid detection method for steel surface defects. *Measurement* **2022**, *187*, 110211. [CrossRef]
49. Gu, X.; Guo, R.; Wang, H. Quality inspection of workpiece camouflage spraying based on improved YOLOv3-tiny. In Proceedings of the 2020 IEEE 6th International Conference on Computer and Communications (ICCC), Chengdu, China, 11–14 December 2020; IEEE: New York, NY, USA, 2020; pp. 1363–1367. [CrossRef]
50. DiCiccio, T.J.; Efron, B. Bootstrap confidence intervals. *Stat. Sci.* **1996**, *11*, 189–228. [CrossRef]
51. Pek, J.; Wong, A.C.M.; Wong, O.C.Y. Confidence intervals for the mean of non-normal distribution: Transform or not to transform. *Open J. Stat.* **2017**, *7*, 405–421. [CrossRef]
52. OpenCV. Template matching. In *OpenCV Documentation Version 3.4.20-dev*. Available online: https://docs.opencv.org/3.4/d4/dc6/tutorial_py_template_matching.html (accessed on 10 March 2023).