applied sciences

MDPI

Article

A Defect Detection Method Based on YOLOv7 for Automated Remanufacturing

Guru Ratan Satsangee , Hamdan Al-Musaibeli and Rafiq Ahmad *

Smart & Sustainable Manufacturing Systems Laboratory (SMART Lab), Department of Mechanical Engineering, University of Alberta, 9211 116 Street NW, Edmonton, AB T6G 2R3, Canada; satsange@ualberta.ca (G.R.S.); almusaib@ualberta.ca (H.A.-M.)

* Correspondence: rafiq.ahmad@ualberta.ca

Abstract: Remanufacturing of mechanical parts has recently gained much attention due to the rapid development of green technologies and sustainability. Recent efforts to automate the inspection step in the remanufacturing process using artificial intelligence are noticeable. In this step, a visual inspection of the end-of-life (EOL) parts is carried out to detect defective regions for restoration. This operation relates to the object detection process, a typical computer vision task. Many researchers have adopted well-known deep-learning models for the detection of damage. A common technique in the object detection field is transfer learning, where general object detectors are adopted for specific tasks such as metal surface defect detection. One open-sourced model, YOLOv7, is known for real-time object detection, high accuracy, and optimal scaling. In this work, an investigation into the YOLOv7 behavior on various public metal surface defect datasets, including NEU-DET, NRSD, and KolektorSDD2, is conducted. A case study validation is also included to demonstrate the model's application in an industrial setting. The tiny variant of the YOLOv7 model showed the best performance on the NEU-DET dataset with a 73.9% mAP (mean average precision) and 103 FPS (frames per second) in inference. For the NRSD dataset, the model's base variant resulted in 88.5% for object detection and semantic segmentation inferences. In addition, the model achieved 65% accuracy when testing on the KolektorSDD2 dataset. Further, the results are studied and compared with some of the existing defect detection models. Moreover, the segmentation performance of the model was also reported.

Keywords: YOLOv7; metal defect; defect detection; defect segmentation

check for updates

Citation: Satsangee, G.R.; Al-Musaibeli, H.; Ahmad, R. A Defect Detection Method Based on YOLOv7 for Automated Remanufacturing. *Appl. Sci.* **2024**, *14*, 5503. https://doi.org/10.3390/app14135503

Academic Editors: Vytautas Bucinskas, Andrius Dzedzickis and Janis Arents

Received: 28 May 2024
Revised: 20 June 2024
Accepted: 21 June 2024
Published: 25 June 2024

1. Introduction

Remanufacturing is gaining more attention for its significant impact on establishing a sustainable circular economy [1]. This increase in interest is mainly due to its capacity for resource efficiency and waste reduction. A critical aspect of remanufacturing is the difficulty in aligning production planning and control with remanufacturing processes. Guide [2] extensively discussed the research needs in this domain, highlighting the gap between industry practice and theoretical development. Additionally, the development of a remanufacturing supply chain management system, as exemplified in the case study by Zhu and Tian [3], further demonstrates the practical applications and benefits of remanufacturing in the industry. The evolution of remanufacturing processes is further emphasized by Tolio et al. [4] and Caterino et al. [5], who highlighted the integration of cutting-edge systems like cloud technologies in remanufacturing.

One crucial step in remanufacturing involves inspecting end-of-life (EOL) products, which is mostly carried out manually by a skilled worker [6]. However, with promising achievements in computer vision applications, scientists are now focusing on automating this remanufacturing step by adopting intelligent defect detection approaches to recognize faulty regions on EOL parts that require restoration [7–9].

Defect detection on metal surfaces [10] is an important area of research that is gaining more attention from the scientific community as we move into a sustainable environment. Defects on metal surfaces occur in several types, including cracks, scratches, inclusions, corrosion, spots, patches, etc. [11]. Such defects can affect the quality and performance of the product. Chen et al. [12] conducted an extensive exploration of traditional and deep learning-based methods for surface defect detection, underscoring the evolution from basic feature analysis to sophisticated neural architectures capable of handling complex defect patterns. Furthermore, Jin et al. [13] extended the discussion to the application of machine learning in solid mechanics, suggesting a broader context where these computational models contribute to predictive maintenance and quality assurance in manufacturing processes. Thus, there have been many efforts to develop and adopt different models for efficient and accurate detection of such defects.

Some researchers have focused on developing novel deep learning architecture targeting a specific environment and dataset [14–18]. Others adopted and fine-tuned some of the existing models trained and developed for a general object detection domain [7–9,19–22]. A recent approach involves improving a well-performed existing model utilizing hybrid methods [23–26]. Therefore, a behavioral study of the state-of-the-art object detection model on detecting defects on metal surfaces is important for further decision support in the selection of the repair process. Thus, this paper focuses on investigating the state-of-the-art object detection model for the defect detection task.

Many novel approaches are based on convolutional neural networks (CNNs). CNNs are robust networks used for extracting embedded features for images, such as corners, edges, etc., which makes them widely adopted in the object detection task [27]. Tao et al. [14] developed a detection pipeline based on cascaded autoencoder (CASAE) and CNN networks. This architecture uses CASAE to localize and segment the defect, whereas CNN is used to classify the defect type. Moreover, Parvez et al. [15] and Han et al. [16] designed similar networks that used CNN for feature extraction and a fully connected neural network (FCNN) for the classification step. Both focused on detecting defects in additively manufactured parts, which include cracks, porosity, and lack of fusion. In addition, Xu et al. [17] developed a novel self-supervised efficient defect detector (SEDD) that focuses on eliminating the annotation step of the data by using a homographic enhancement method. They also designed a custom detector based on depth-wise convolution layers and an attention module to enhance performance and accuracy.

Another approach is to utilize state-of-the-art object detection detectors for defect detection applications. This approach is known as transfer learning, where the model is trained on a new dataset to perform a related task using initial weights [28]. Zheng et al. [7,19] adopted mask region-based CNN (Mask RCNN) architecture for detecting and segmenting the damaged area on metal parts for further repair applications. In addition, Zheng et al. [20] investigated the performance of Faster R-CNN, YOLOv3, and RetinaNet architectures on rail crack detection employing knowledge transfer of the COCO dataset [29]. Furthermore, Konovalenko et al. [21] and Litvintseva et al. [22] based their detection model on the U-Net architecture for defect detection. In [8,9], Imam et al. leveraged one of the most known two-stage models—Faster R-CNN using the transfer learning concept (using the COCO dataset) to detect and localize steel parts' wear for a remanufacturing task.

A recent work by Li et al. [23] proposed a new method that leverages and enhances CSPDarknet53 architecture by integrating a multi-head self-attention block and using it as the backbone of their detector. They also adopted some simple yet effective techniques to enhance the model's performance, such as augmentation and grayscale filtering. Moreover, Pan et al. [24] integrated a dual attention module with DeepLabv3+ architecture to detect and segment metal defects. Furthermore, Wang et al. [25] developed a new module that follows a similar pipeline to ResNet based on Transformers and CNNs to retain both local and global information. Moreover, Gao et al. [26] adopted the Swin Transformer with a variant shift window called Cas-VSwin Transformer as the backbone network for better performance on the defect detection task. In addition, they used the feature pyramid network (FPN) as the neck, whereas the Cascade Mask network is the head of the architecture.

One feature of YOLO detectors is real-time inference. They gained a lot of attention on the object detection task. With the speed requirement for the defect inspection process in production lines, many scientists have investigated and adopted YOLO for the defect detection problem. In addition, the YOLO series provides the lowest processing time compared to all other state-of-the-art detectors. However, higher detection accuracy is achieved with enhanced versions of the YOLO series by integrating it with other networks, such as self-attention mechanisms. Li et al. [30] developed an improved version of the YOLO network using only CNN layers. This method performed well; however, no comparison with the standard YOLO network was conducted.

Furthermore, Kou et al. [31] proposed a new detector based on YOLOv3 architecture with an anchor-free feature selection method and custom dense convolution blocks. This improves the training process and inference accuracy by about 26.7% on the NEU-DET dataset. In addition, Xu et al. [32] modified the YOLOv3 network to improve accuracy by focusing on extracting more features of small defects using a new scale feature layer. With this, an improvement of 4.6% in precision is gained on the Tianchi dataset. On the other hand, Guo et al. [33] introduced an improved architecture of YOLOv5 with a Transformer encoder as the backbone of the network to integrate more global information into the model. This model was tested with the NEU-DET dataset and achieved 75.2% mean average precision (mAP) on inference. Saiz et al. [34] also proposed combining YOLOv5 and DeepLabV3+ models, which provides stable and reliable defect detection in components, overcoming the limitations of traditional methods. This ensemble approach, tailored to specific use cases, achieved full accuracy in overall performance testing, proving highly effective for practical applications in the manufacturing industry.

It is essential to acknowledge the complexities introduced by the diverse range of defect types, such as scratches, dents, and corrosion, each varying significantly in shape, size, and visibility [14]. Traditional computer vision techniques, while less computationally demanding, often fall short in flexibility and adaptability, struggling with high noise levels and variability in defect manifestations, which are common in industrial settings [35]. Conversely, deep learning models, despite their superior performance in learning from large datasets and generalizing across different defects, require extensive computational resources and substantial amounts of labeled data, which complicates their application in real-life industrial scenarios [36]. One of the most promising solutions for industries is the YOLOv7 object detection model, which aims to bridge these gaps by enhancing detection accuracy while meeting the operational demands of real-time industrial applications [37].

YOLOv7 is the current state-of-the-art for real-time object detection tasks in terms of both speed and accuracy [37]. However, this benchmark is based on the Microsoft COCO dataset. A benchmark of YOLOv7 on the defect datasets is essential to observe and investigate its performance on metal defect detection tasks. This paper focuses on adopting the state-of-the-art model, YOLOv7, for detecting defects and damages on metal surfaces and investigating the model's behavior on some public datasets of metal defects. This can also be used as a reference for future comparisons of the enhanced models based on YOLOv7 with the standard. Furthermore, a comparison of the existing defect detection models with the YOLOv7 is conducted and studied. A case study validation is also conducted on synthetic data inspired by an industrial setting, demonstrating the real-world application and efficacy of the YOLOv7 model in identifying and classifying defects.

2. YOLOv7 Architecture

YOLOv7 is the latest improved model of the YOLO series. YOLO models, including YOLOv7, are a type of single-stage framework that contains three main components, named backbone, neck, and head [38–44], as shown in Figure 1. The backbone extracts feature maps of an image and transfers them to the neck layers. Those maps are combined, fused, and passed to the next layers. Then, the head network predicts the objects' bounding boxes and their classes. Unlike the two-stage models, the YOLO series, a single-stage model, reconsidered the object detection task as a regression problem instead of a classification problem, which is the main feature of real-time detection algorithms [45].

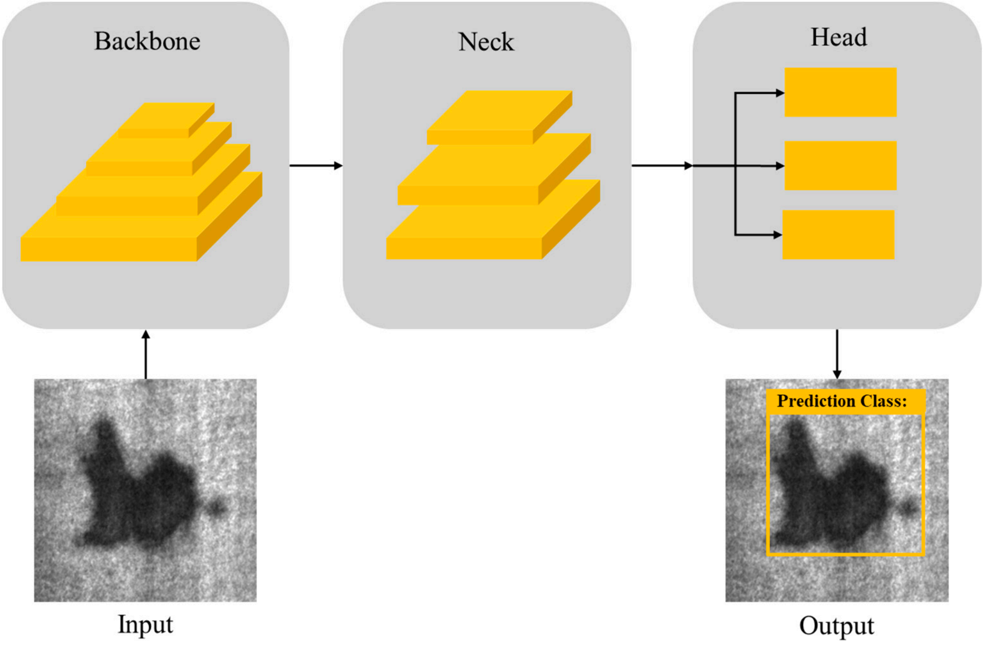


Figure 1. YOLOv7 architecture design.

The efficiency of the CNNs in the backbone network is important to enhance the inference process. Thus, the author of YOLOv7 proposed an enhanced method based on an efficient layer aggregation (ELAN) network named Extended-ELAN (E-ELAN). This method improves the ELAN architecture to boost the learning ability of a scaled network without disturbing or changing the original gradient propagation path. E-ELAN has a stronger learning ability for various features.

Furthermore, YOLOv7 comes with a new method of scaling for concatenation-based models. The proposed method, named corresponding compound model scaling, addresses the issue of a larger width output of the computational block when depth scaling is performed on the architecture. With the proposed method, the depth of the concatenation-based model is scaled directly; however, the width of transition layers is scaled with a corresponding factor calculated from the change of the output width of the block.

Moreover, several techniques have been introduced in the YOLOv7 to improve the model inference accuracy while maintaining a low training cost. Those strategies are called bags of freebies (BoF), including planned re-parameterization and dynamic label assignment. After a deep investigation of the re-parametrized convolution behavior when combined with various networks, the author showed an increase in the model's accuracy when using the RepConv without identity connection (RepConvN). Further, in supervised deep learning techniques, the head that represents the final prediction of the model is called the lead head, whereas the auxiliary head is the head that is used to assist the lead head. Previously, both heads were independent of each other, and their prediction and the ground truth were used as soft labels for label assignment. However, YOLOv7 proposed a new method for lead-dependent label assignment. Two types of label assigners were developed with the YOLOv7. One is lead head guided, where the soft label is mainly generated from the leader's head and ground truth. Another is coarse-to-fine lead head-guided, where two different soft labels are produced, including fine and coarse labels. The fine label is the same as the one generated in the lead head guided assigner; however, the coarse label is generated with relaxed rules on the positive sample assignment process.

Other BoF techniques are also adopted and used in YOLOv7, such as batch normalization, implicit knowledge, and the exponential moving average (EMA) model. Normalization of the training batch, by integrating the mean and variance of the data to the bias and weight of the convolutional layer, is proved to directly affect the training process by utilizing a higher training rate and faster convergence [46]. Another technique is the implicit knowledge, adopted from the YOLOR [44], computed as a vector in the inference stage of YOLOv7, improving the prediction accuracy in previous versions. Lastly, adopting the EMA model as the final inference model in YOLOv7 has improved the inference accuracy.

YOLOv7 outperforms all existing models in the object detection task in terms of both speed and accuracy [37]. According to the authors, the focus of YOLOv7 has been to optimize the training process for enhanced detection accuracy and speed, as well as to improve the inference process. This optimization includes the reduction of model training parameters and the enhancement of the learning process.

In contrast to the evaluation performed in [37], which establishes the general superiority of YOLOv7 in object detection, this paper uniquely contributes to the field by explicitly investigating the performance of YOLOv7 in detecting defects on metal surfaces. While previous benchmarks, such as the one on the Microsoft COCO dataset, provide a broad understanding of YOLOv7's capabilities, this work offers a specialized assessment of its applicability and efficiency in the context of metal surface defect detection. This evaluation is essential, as the characteristics and requirements of defect detection on metal surfaces differ significantly from the more general object detection tasks. Therefore, this analysis reinforces the findings regarding YOLOv7's overall performance and extends its utility to the specific domain of defect detection, providing valuable insights for industrial applications.

3. Training and Results

3.1. Datasets

Since YOLOv7 was trained and tested on the Microsoft COCO dataset, the model's behavior on the metal defect datasets has not yet been studied by researchers. Therefore, this paper investigates the behavior of the YOLOv7 model on metal defect detection tasks. An optimal approach for this study is to use public datasets for standard benchmarks and replication of the same results in the future. In addition, a comparison of the YOLOv7 performance with previously reported results is conducted. A list of the available public datasets is mentioned in Table 1.

Table 1. Public metal defect datasets.

Dataset	Set Size	Raw Image Size (Pixels)	# of Classes
Severstal defect dataset [47]	18,074	1600 × 256	4
No-service rail surface defect (NRSD) [48]	4101	600 × 600	1
Kolektor surface-defect dataset 2 (KSDD2) [49]	3335	230 × 630	1
DAGM 2007 [50]	2300	512 × 512	10
GC10 defect dataset (GC10-DET) [51]	2294	415 × 416	10
Northeastern University defect dataset (NEU-DET) [52]	1800	200 × 200	6
Ball screw drive surface defect dataset (BSData) [53]	1104	1130 × 460	1
Kolektor surface-defect dataset (KSDD) [54]	399	500 × (1240–1270)	1
Rail surface discrete defect (RSDD) [55]	167	various	1

There are many public datasets available for the metal defect detection task; however, investigating the YOLOv7 model performance on all of them is time and resource-consuming and adds a minimal contribution to the study. Thus, three of the easily available and frequently cited datasets listed in Table 1, are selected to perform this investigation and obtain a general performance of the YOLOv7 model on the defect detection task. Those are NEU-DET, NRSD, and KolektorSDD2 datasets. Samples of the annotated datasets are shown in Figures 2–4.

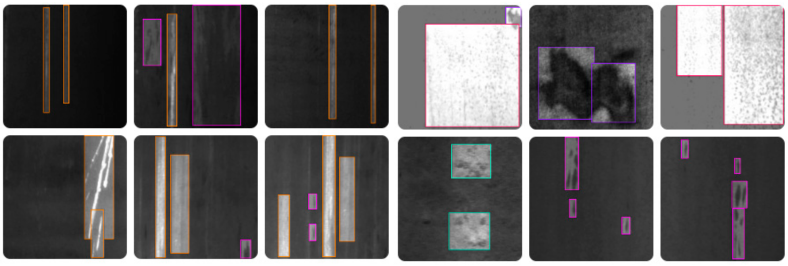


Figure 2. Annotated samples of the NEU-DET dataset.

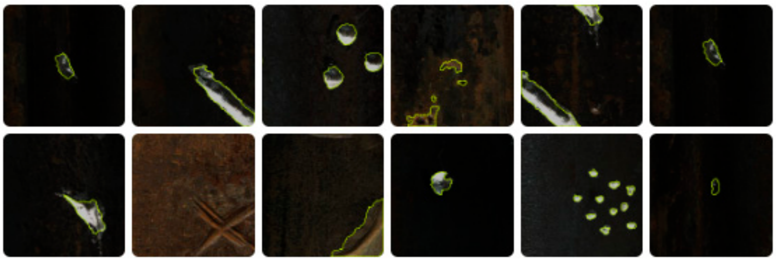


Figure 3. Annotated samples of the NRSD dataset.

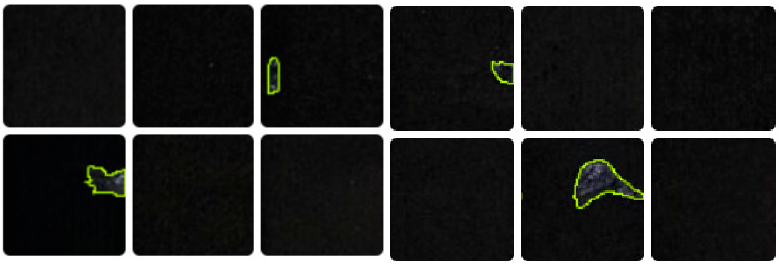


Figure 4. Annotated samples of the Kolektor 2 dataset.

The NEU-Det dataset contains grayscale images collected from real-life industrial settings, specifically from the inspection of metal surfaces in manufacturing and remanufacturing environments. These images are representative of common defects found in such contexts, including but not limited to cracks, scratches, inclusions, corrosion spots, and other anomalies typically encountered IN metal surfaces during production and post-production processes. In addition, the NEU-Det dataset is annotated for an object detection task meaning only bounding boxes of the defect exist, whereas NRSD and KSDD2 are annotated for both object detection and segmentation tasks containing bounding boxes and masks. Furthermore, NRSD and KSDD2 consist of colored (i.e., RBG) images of planned and unplanned defects. Furthermore, it is important to note that the NRSD dataset contains some synthetic segmentation of the collected images generated by MCnet. Therefore, annotation is not accurate and might cause difficulty for the model to perform. Further data preprocessing is required to address these annotation inaccuracies.

3.2. Training Environment

The model was trained and tested locally on a machine with the specifications listed in Table 2. Those specifications are considered at the high end of the time performing this study.

Table 2. Training machine specifications.

Property	Value
CPU	AMD Ryzen Threadripper 3970X 32-Core
GPU	NVIDIA GeForce RTX 3090/24 GB
CUDA cores/version	10,496/11.8
Operating system	Windows Server 2019
RAM	128 GB
PyTorch	1.10.1

3.3. Training Parameters

To maintain a standard benchmark with the original results of the YOLOv7 models, this work uses similar training parameters as studied by Wang et al. [37] with an increased training iteration for the larger variants of the YOLOv7 model, i.e., yolov7-d6, yolov7-e6, and yolov7-e6e. There are three different hyper-parameter settings depending on the model size, including small, medium, and larger. Furthermore, each dataset has a different set size and image resolution; as a result, training hyper-parameters are slightly different for each dataset. A general range of those parameters are listed in Table 3. In addition, each dataset is split into three sets: 70% training, 20% validation, and 10% testing.

Table 3. Training parameters.

Parameter	Value	Parameter	Value
Learning rate	0.001	Batch Size	8–32
Momentum	0.937	Image Size	Depends on dataset
Weight decay	0.0005	Epochs	100–300

3.4. Evaluation Metrics

In this work, standard evaluation metrics are adopted, the same as in YOLOv7, to study and compare the performance of the different variants of the YOLOv7 models on different metal defect datasets. Traditionally, the mean average precision (mAP), which is the area under the precision and recall curve calculated using (1), is calculated at a 0.5 threshold of the intersection over union (IoU) (mAP_0.5). However, a recent trend in the research field is to compute mAP over multiple IoU values following the COCO interpretation from 0.5 to 0.95 with a step of 0.05 (mAP_0.5:0.95). This has proven to affect the model with a better localization reward. However, some of the recent studies in the defect detection field still use the mAP_0.5; thus, both metrics will be reported in this study. In addition, inference speed is also observed for each dataset.

$$AP = \int_0^1 (precision \times recall)\, d(recall) \tag{1}$$

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

where *TP*, *FP*, and *FN* are the true positive, false positive, and false negative of the bounding box predictions, respectively.

3.5. Training Results

The YOLOv7 model was trained on each of the selected datasets in Section 3.1. to measure its performance in detecting defects on metal surfaces. Since the NEU-Det dataset was annotated for an object detection task meaning, Section 1 of this study will focus on investigating the behavior of each variant of the model on this dataset. The remaining investigation of this study is to analyze and report the performance of the base variant of the YOLOv7 model on the metal defect detection and segmentation task. Moreover, this work used the initial weights of the trained YOLOv7 model on the Microsoft COCO dataset. This is to leverage and transfer the knowledge from the previous object detection task and facilitate the training process for faster convergence.

3.5.1. NEU-DET

In this study, each model variant was trained and tested on this dataset to observe a wider behavior of the model. The validation and inference results are shown in Table 4. There are three main evaluation metrics presented in the table. AP^{test}, AP^{val}, and $AP_{50}{}^{test}$ are mAP_0.5:0.95 for the testing set, mAP_0.5:0.95 for the validating set, and mAP_0.5 for the testing set, respectively. Those parameters are selected for better visualization of the model's performance. As shown in the table, the main difference between each variant is the model size, increasing as going down the table. The larger the model size, the more training parameters it has, which increases the learning knowledge and training time.

Table 4. Performance of each variant of YOLOv7 on NEU-DET metal defect datasets.

YOLOv7 Variant	#Params	Image Size	$FPS^{RTX\ 3090}$	AP^{test}/AP^{val}	$AP_{50}{}^{test}$
Tiny	6 M	224	103	37.0%/35.6%	73.9%
Base	37 M	224	78	37.1%/35.4%	73.9%
X	70.8 M	224	63	30.2%/30.3%	65.8%
W6	81 M	448	61	31.7%/30.5%	69.3%
E6	110 M	448	45	31.1%/29.8%	67.2%
D6	153 M	448	40	33.8%/32.6%	70.9%
E6E	164 M	704	30	36.0%/31.7%	73.3%

Furthermore, the smallest variant of the YOLOv7 model took about 84 min to complete 150 epochs of training, whereas the largest version took four times the smallest one. The table shows that the YOLOv7 achieved about 73.9% mAP with a 0.5 threshold on the testing set. However, as expected, a lower accuracy was observed with an interval threshold of (0.5–0.95). Although each variant was trained in slightly different parameters, the reported results are all for 150 training epochs. In Figure 5, the training process curve of each variant is plotted to represent the mAP_0.5. As shown in this figure, the small-sized models tend to converge faster compared to the larger ones due to the number of parameters that require optimization. It is also worth noticing that smaller models have a smoother learning curve, which might be due to the loss of some information in the images when resizing, especially for the small objects. In addition, a smaller batch size might reduce the sharp fluctuation in the learning curve. Figure 6 shows the mAP_0.5 of the testing set with its trend. The performance trend of the YOLOv7 model variants represents a parabolic trend where the performance decreases in the middle. This could be due to the increase in the image size and the model, which led to more knowledge gained.

This dataset is public and has been used in serval studies. Furthermore, comparing the YOLOv7 results with other previously reported results is valuable. Lv et al. [51] reported a 72.2% in mAP_0.5 on their proposed method based on EDDN. Their model performed slightly better in the pitted surface and scratches classes but worse in detecting cracks. Furthermore, Guo et al. [33] reported better results with 75.2% of mAP_0.5 of their proposed model based on an improved YOLOv5 an increase of about 1.3%. Moreover, a recent model proposed by Gao et al. [26], based on Transformers, has achieved higher results in the

NEU-DET dataset with 80.5% mAP_0.5. It is worth noting that integrating YOLO models with Transformer networks enhances the model performance, as shown; thus, it is worth exploring this approach to further increase detection accuracy. Lastly, the base variant of the YOLOv7 model provides the right balance between accuracy and speed for real-time defect detection.

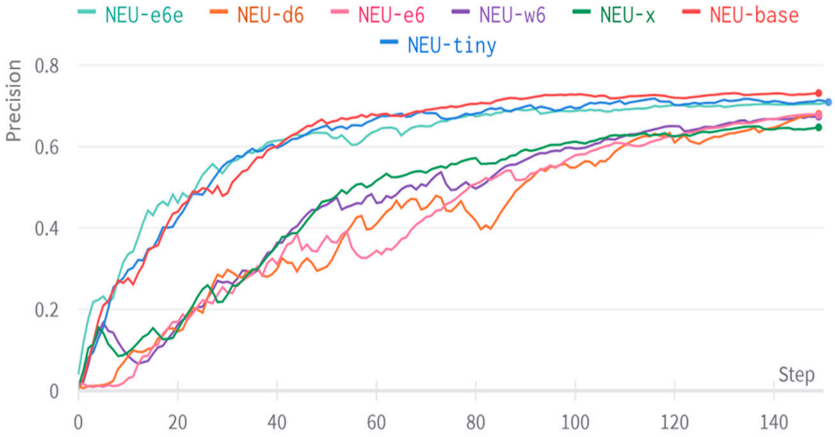


Figure 5. YOLOv7 variants' training process on the NEU-DET dataset (mAP_0.5).

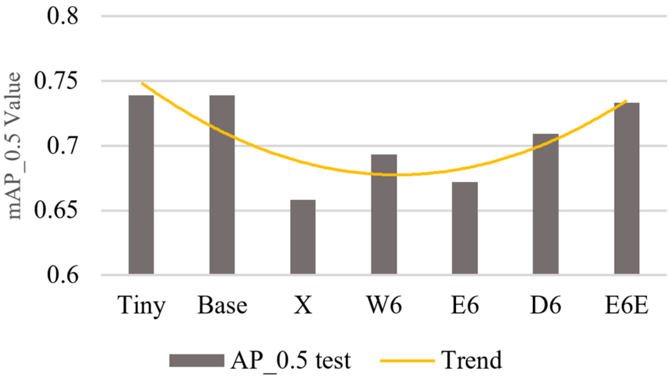


Figure 6. YOLOv7 variants' testing performance on the NEU-DET dataset.

Looking further into the base variant results, the model struggles to detect cracks and rolled-in scale defects with about 19% of mAP, as reported in Table 5 and shown in Figure 7. This might be due to the nature of those types of defects existing with minimal features and are less distinguishable with the metal surface compared to other types of defects. In addition, the ratio of defect to background pixels in those types is usually small compared to other types. In contrast, patch defect provides unique features with a high defect/background ratio leading to better results of about 54% mAP.

Table 5. YOLOv7-based variant model performance per class.

Class	AP_{50}	mAP
Crack	54.5%	19.0%
Inclusion	81.9%	40.8%
Patch	92.4%	54.2%
Pitted surface	79.7%	39.8%
Rolled-in scale	55.1%	18.5%
Scratch	80.5%	40.2%

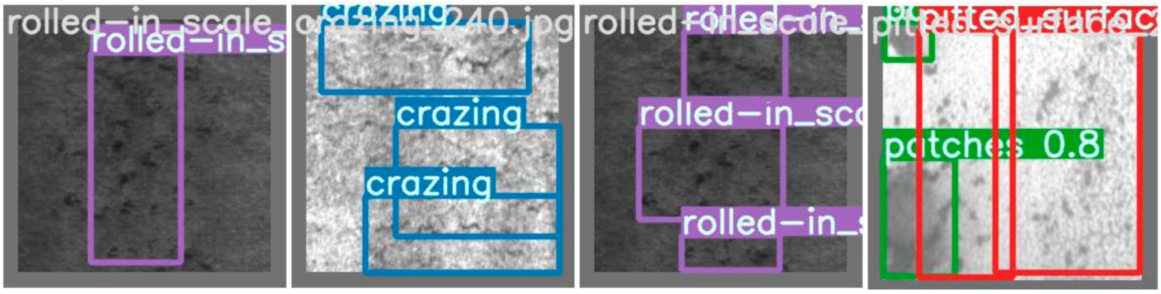
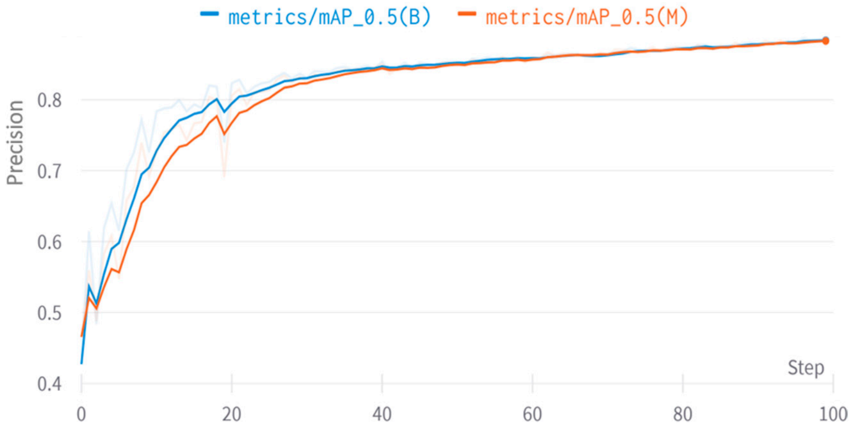


Figure 7. Samples of the NEU-DET dataset inference results.

3.5.2. NRSD

For this dataset, the base variant of the YOLOv7 model was selected to investigate its instance segmentation performance. Although this dataset contains multiple types of defects, including scratches, wear, welding spots, etc., it was annotated with one class which is a defect. Furthermore, this work reports both object box and mask results. The dataset images were all resized to 640 × 640 for the training process. The model was trained for 100 epochs and 24 batch sizes, and the same parameters mentioned in Table 3 were used. As shown in Figure 8, the YOLOv7 model performed better with the NRSD dataset than the NEU-DET for many reasons, including RGB information, higher quality images, large objects, higher defect/background ratio, and larger dataset size. The model has a similar performance for detecting the bounding box of the defects with about 88.5% for both metrics (mAP_0.5 and mAP_0.5:0.95). However, the segmentation task has a slightly worse accuracy, where it achieved 69.6% mAP_0.5:0.95 in detecting masks. The similarity of the defects to the background might affect the segmentation results as shown in the testing results in Figure 9. Furthermore, the segmentation training process coverage is slower than the object detection process due to the larger knowledge that needs to be learned. Thus, increasing training epochs might increase segmentation accuracy. In addition, the inference process is observed to take about 18 ms per image on RTX 3090 GPU that about 55 fps. Some of the testing results on NRSD dataset are shown in Figure 10. In contrast with previously reported results, Li et al. [23] proposed a detection model that accomplished 81.09% of mAP_0.5, which is lower by about 6.91% compared to the YOLOv7 model.



(**a**)

Figure 8. *Cont.*

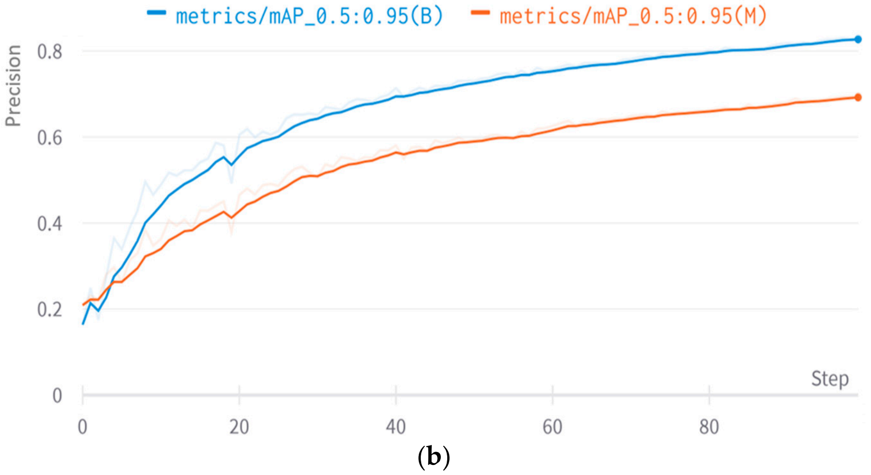


Figure 8. YOLOv7-based variant model training processing performance on the NRSD dataset (B: box, M: mask). (**a**) mAP_0.5; (**b**) mAP_0.5:0.95.

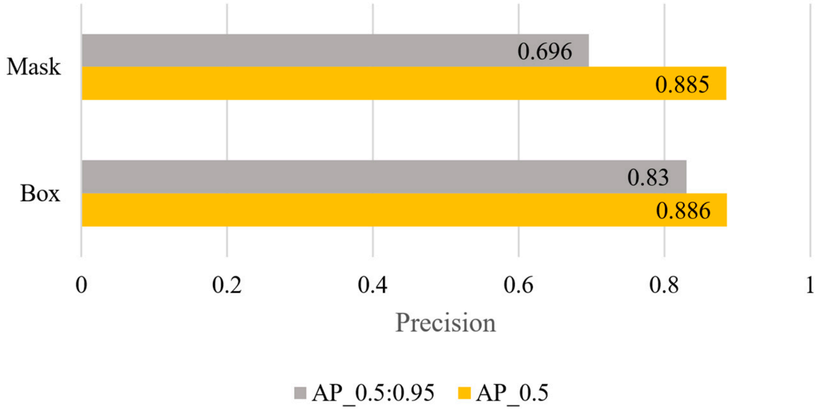


Figure 9. YOLOv7-based variant model inference results.

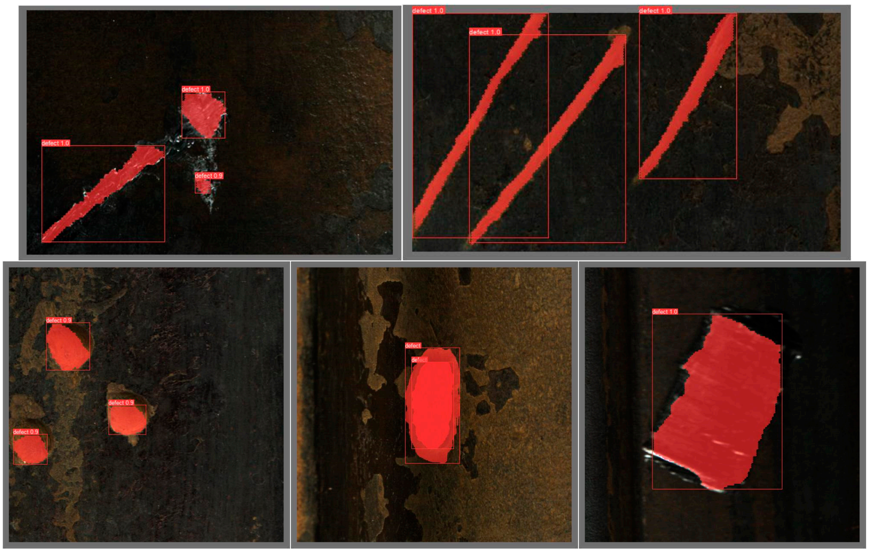
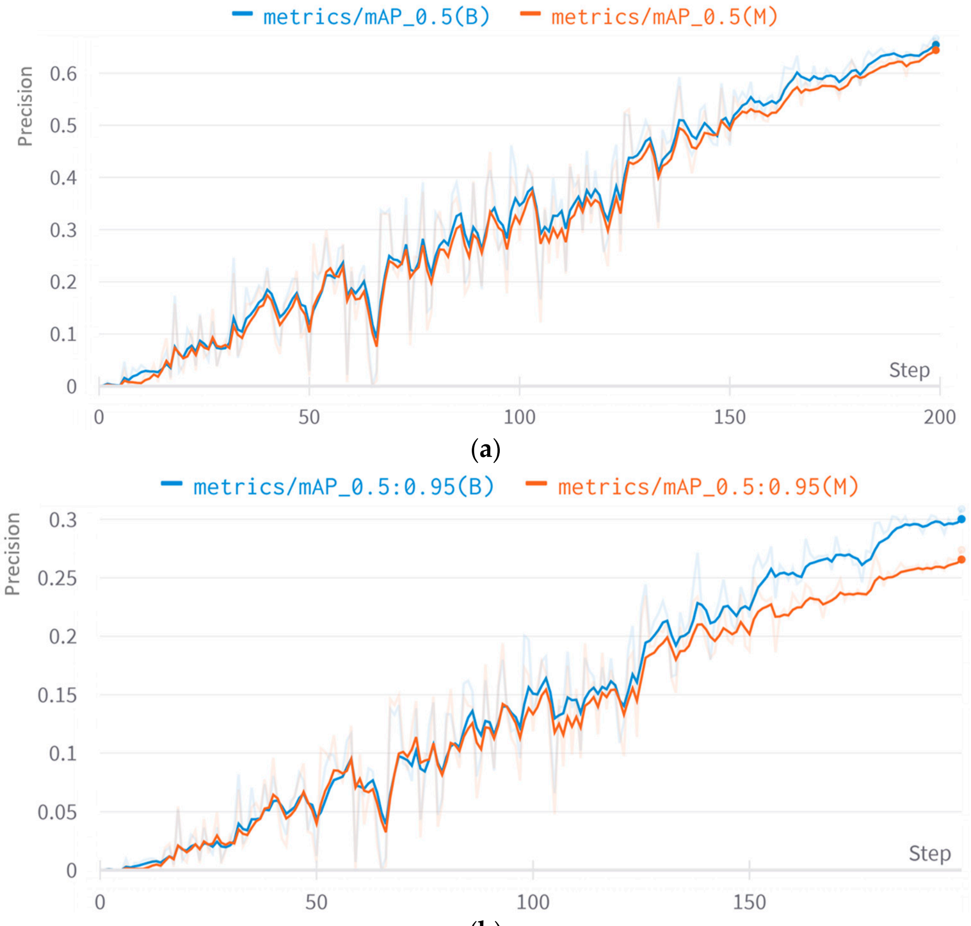


Figure 10. Sample results of the model prediction for the NRSD dataset.

3.5.3. KSDD2

In this challenging dataset, where samples containing defects are only about 10% of the total samples, training results are shown in Figure 11. As the NRSD dataset, the base model variant was selected and trained with similar parameters as with NEU-DET mentioned in Table 3. In addition, images are resized to 640 × 640 to reduce training time. Furthermore, the model was trained for 200 epochs since this dataset has fewer true positive samples, which is expected to slow the learning process, as shown in Figure 11. The training process results plotted in Figure 11 show a difficult learning curve due to serval reasons including a small number of defect samples, and a similar texture of defect with the background. Some of the testing results are shown in Figure 12. For the mAP with a 0.5 IoU threshold, the model performance with about 65% for both the bounding box and mask. However, the accuracy is reduced dramatically for the mAP over the range (0.5–0.95) IoU threshold with about 30% for the bounding box and 26% for detecting masks. This is due to smaller portions of the defective areas being detected that are ignored when the threshold is larger. For this dataset, Jakob et al. [49] introduce a weakly supervised model that achieved a 73.3% mAP_0.5 which is about 8.3% better than the YOLOv7 model in segmentation.



(**a**)

(**b**)

Figure 11. YOLOv7-based variant model training processing performance on the KSDD2 dataset (B: box, M: mask). (**a**) mAP_0.5; (**b**) mAP_0.5:0.95.

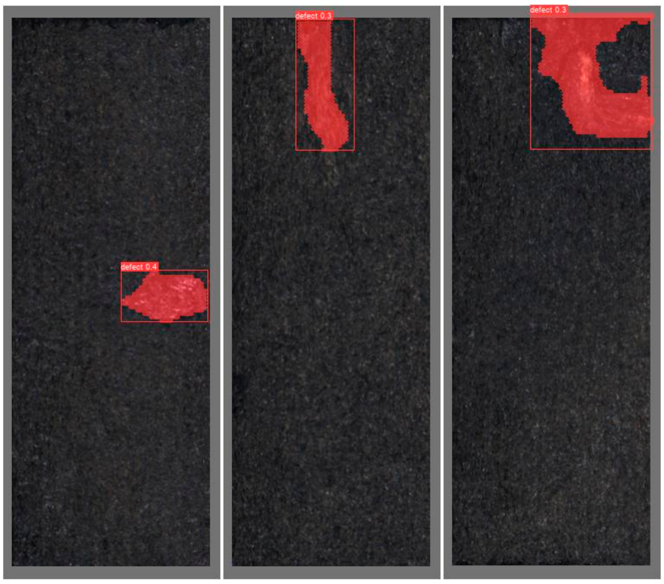


Figure 12. Sample results of the model prediction for the KSDD2 dataset.

Table 6 provides a comparison of the YOLOv7 defect detection models used in this study across different datasets, along with some other studies, highlighting the accuracy metrics for each model. The results illustrate the strengths and limitations of each approach in identifying defects in industrial contexts. YOLOv7 variants show strong performance on the NRSD dataset but have lower performance with the KolektorSDD2 dataset, indicating variability in their effectiveness based on dataset characteristics. ResNet50 (Faster R-CNN) and Mask R-CNN demonstrate high accuracy on custom datasets, suggesting their suitability for specific applications. The Cas-VSwin Transformer balances high accuracy in both box and mask predictions, highlighting the importance of hybrid methods that combine YOLO with Swin Transformers, making it more suitable for real-time applications.

Table 6. Comparative performances of defect detection models.

Method	Dataset	Accuracy
YOLOv7 variants	NEU-DET	65–73% mAP_0.5
YOLOv7 (base)	NRSD	88.5% mAP_0.5
YOLOv7 (base)	KolektorSDD2	65% mAP_0.5
ResNet50 (Faster R-CNN) [9]	Custom (fixed bends)	88.7% mAP
Cas-VSwin Transformer [26]	Private dataset	82.3% AP (Box)/80.2% AP (Mask)
Mask R-CNN [56]	Custom insulator dataset	87.5% mAP
Faster R-CNN [57]	Aluminum defect dataset	78.8% mAP

There is potential for further improvements in the average prediction precision of this study through advanced data augmentation techniques. Techniques such as geometric transformations (rotations, translations, scaling, and flips) [58], photometric adjustments (altering brightness and contrast) [59], and the addition of synthetic noise can significantly enhance the model's robustness. Additionally, more sophisticated methods like generative adversarial networks (GANs) can be used to generate synthetic data that mimics real-world variations in defects [60]. Implementing these data augmentation methods is expected to improve the robustness of the YOLOv7 model and its precision in detecting subtle and complex defects.

4. Case Study Validation

To further validate the efficacy of the YOLOv7-based defect detection model, a case study was conducted on a damaged cylindrical component. This study primarily focused on the identification of wear-type defects, which are prevalent and repairable in laser-cladding processes. The NRSD dataset, notable for its high-resolution RGB images, variety of defect types, and texture that mimics real damaged components, was employed for model training.

In this case study, a virtual replica of a damaged workpiece was created using Blender software version 3.0 (Figure 13), incorporating a metallic texture inspired by real-life damaged samples (Figure 14). This virtual replica was used to ensure controlled testing conditions and simulate specific defect types uniformly. A single wear defect, representative of common wear and tear such as dents, was introduced onto this virtual part. Subsequent image captures of this damaged part, aided by a 2D camera and a pixel stitching algorithm, facilitated the creation of a comprehensive image for analysis (Figure 15). The pre-trained YOLOv7 model, applying bounding box techniques, successfully detected these synthetic defects with an 86% accuracy on mAP. This finding underscores the potential of the model, although the expansion of validation samples and further real-life experiments are necessary for comprehensive performance verification.

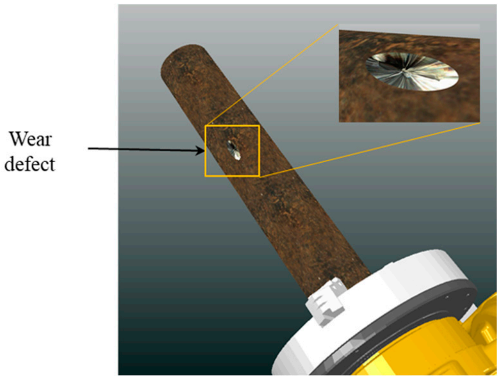


Figure 13. Virtual Model of the damaged sample.

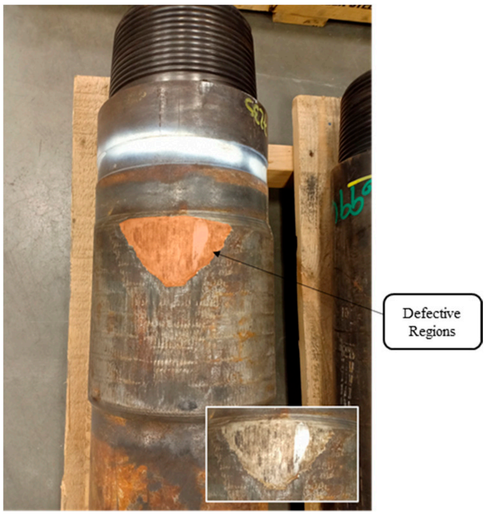


Figure 14. The actual damaged sample repaired via a laser-cladding process.

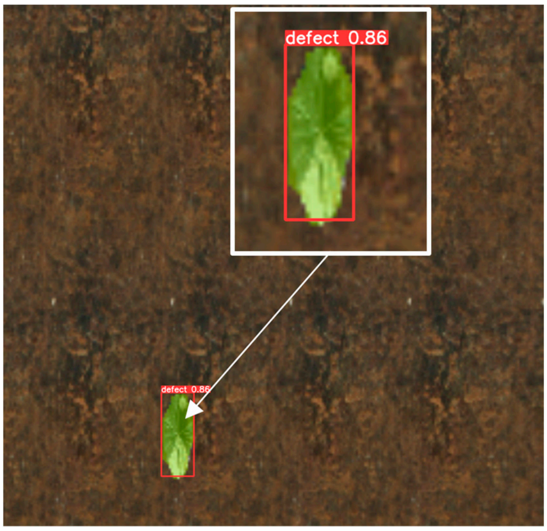


Figure 15. Defect detection on the generated stitched map.

5. Conclusions

Deep learning architectures have shown promising results in surface defect detection. One of those is the YOLOv7 model, which is state-of-the-art in the object detection field and outperforms real-time detectors in terms of speed and accuracy. In this study, an investigation of the YOLOv7 model is conducted on a metal defect detection task using a publicly available dataset. Some of the datasets are selected for this study according to specific criteria. The selected datasets are NEU-DET, NRSD, and KolektorSDD2. Further, each variant of the YOLOv7 model is studied on the NEU-DET. The mAP_0.5 achieved on this dataset was about 65–73%, with the base model coming with the optimal balance between accuracy and speed. In addition, object detection and semantic segmentation of the base variant of the YOLOv7 model were tested on the NRSD dataset. The YOLOv7 model achieved a higher accuracy of about 88% on mAP_0.5 with this dataset for its larger set size, higher resolution, and larger defect/background ratio. In the last dataset, the model was trained for some extra time to compensate for the fewer defect samples; however, the model achieved the worst accuracy among all tested datasets. This is due to the higher similarity of the defect to the background texture. All variants of the model have accomplished a real-time inference process. For industrial applications, deploying the defect detection model requires more fine-tuning to the specific problem for reliable and accurate results. This tunning process is achieved by training the detection model on real-life defect samples. The performance of the detection model is sensitive and biased to the collected dataset, which might raise some challenges in achieving the desired results. For future development of this pipeline, a custom real-life dataset needs to be collected. Further, a comparative analysis with results obtained by Wang et al. [61] underscores the potential advantages of custom modifications to the YOLOv7 model, as their tailored enhancements led to higher detection accuracy of mAP 81.9% on the NEU-DET database. This highlights an avenue for future research to explore and implement specific model adjustments for improved performance in defect detection tasks. In addition, it is important to mention that hybrid methods such as improving YOLO with Swin Transformers also enable achieving higher accuracy results [26]. Adopting pixel-level supervision could also significantly improve the model's precision by allowing it to learn more granular details of defects, such as small cracks and minute surface variations that are often missed by conventional inspection methods [62,63]. This approach could be particularly beneficial for detecting early-stage defects that typically do not affect the larger visual appearance of the metal surface but may lead to significant issues if not addressed promptly. This work can be used as a base for future studies and comparison of the enhanced models based on YOLOv7.

Author Contributions: G.R.S.: writing—original draft, writing—review and editing; H.A.-M.: investigation, formal analysis, writing—original draft; R.A.: supervision, writing—review and editing, project administration, funding acquisition. All authors have read and agreed to the published version of the manuscript.

Funding: We express our gratitude to the Minister of Economic Development, Trade, and Tourism for funding this project through Major Innovation Funds. We also acknowledge NSERC (grant nos. NSERC RGPIN-2017-04516 and NSERC CRDPJ 537378-18) for further funding of this project.

Data Availability Statement: The raw data supporting the conclusions of this article will be made available by the authors on request.

Conflicts of Interest: We declare that the authors have no competing interests, or other interests that might be perceived to influence the results and/or discussion reported in this paper.

References

1. Khan, S.; Haleem, A.; Fatma, N. Effective adoption of remanufacturing practices: A step towards circular economy. *J. Remanuf.* **2022**, *12*, 167–185. [CrossRef]
2. Guide, V.D.R. Production planning and control for remanufacturing: Industry practice and research needs. *J. Oper. Manag.* **2000**, *18*, 467–483. [CrossRef]
3. Zhu, Q.; Tian, Y. Developing a remanufacturing supply chain management system: A case of a successful truck engine remanufacturer in China. *Prod. Plan. Control* **2016**, *27*, 708–716. [CrossRef]
4. Tolio, T.; Bernard, A.; Colledani, M.; Kara, S.; Seliger, G.; Duflou, J.; Battaia, O.; Takata, S. Design, management and control of demanufacturing and remanufacturing systems. *CIRP Ann. Manuf. Technol.* **2017**, *66*, 585–609. [CrossRef]
5. Caterino, M.; Fera, M.; Macchiaroli, R.; Pham, D.T. Cloud remanufacturing: Remanufacturing enhanced through cloud technologies. *J. Manuf. Syst.* **2022**, *64*, 133–148. [CrossRef]
6. Kaiser, J.-P.; Lang, S.; Wurster, M.; Lanza, G. A Concept for Autonomous Quality Control for Core Inspection in Remanufacturing. *Procedia CIRP* **2022**, *105*, 374–379. [CrossRef]
7. Zheng, Y.; Mamledesai, H.; Imam, H.; Ahmad, R. A novel deep learning-based automatic damage detection and localization method for remanufacturing/repair. *Comput. Aided Des. Appl.* **2021**, *18*, 1359–1372. [CrossRef]
8. Imam, H.Z.; Al-Musaibeli, H.; Zheng, Y.; Martinez, P.; Ahmad, R. Vision-based spatial damage localization method for autonomous robotic laser cladding repair processes. *Robot. Comput. Integr. Manuf.* **2023**, *80*, 102452. [CrossRef]
9. Imam, H.Z.; Zheng, Y.; Martinez, P.; Ahmad, R. Vision-Based Damage Localization Method for an Autonomous Robotic Laser Cladding Process. *Procedia CIRP* **2021**, *104*, 827–832. [CrossRef]
10. Yang, J.; Li, S.; Wang, Z.; Dong, H.; Wang, J.; Tang, S. Using deep learning to detect defects in manufacturing: A comprehensive survey and current challenges. *Materials* **2020**, *13*, 5755. [CrossRef]
11. Mordia, R.; Verma, A.K. Visual techniques for defects detection in steel products: A comparative study. *Eng. Fail. Anal.* **2022**, *134*, 106047. [CrossRef]
12. Chen, Y.; Ding, Y.; Zhao, F.; Zhang, E.; Wu, Z.; Shao, L. Surface Defect Detection Methods for Industrial Products: A Review. *Appl. Sci.* **2021**, *11*, 7657. [CrossRef]
13. Jin, H.; Zhang, E.; Espinosa, H.D. Recent Advances and Applications of Machine Learning in Experimental Solid Mechanics: A Review. *Appl. Mech. Rev.* **2023**, *75*, 061001. [CrossRef]
14. Tao, X.; Zhang, D.; Ma, W.; Liu, X.; Xu, D. Automatic metallic surface defect detection and recognition with convolutional neural networks. *Appl. Sci.* **2018**, *8*, 1575. [CrossRef]
15. Parvez, M.M.; Rahman, M.F.; Galib, S.M.; Liou, F. A Convolutional Neural Network (Cnn) for Defect Detection of Additively Manufactured Parts. In *ASME International Mechanical Engineering Congress and Exposition*; American Society of Mechanical Engineers: New York, NY, USA, 2021; Volume 85550, p. V02AT02A010. [CrossRef]
16. Han, F.; Liu, S.; Liu, S.; Zou, J.; Ai, Y.; Xu, C. Defect detection: Defect Classification and Localization for Additive Manufacturing using Deep Learning Method. In Proceedings of the 2020 IEEE 21st International Conference on Electronic Packaging Technology (ICEPT), Guangzhou, China, 12–15 August 2020; pp. 1–4. [CrossRef]
17. Xu, R.; Hao, R.; Huang, B. Efficient surface defect detection using self-supervised learning strategy and segmentation network. *Adv. Eng. Inform.* **2022**, *52*, 101566. [CrossRef]
18. Wu, H.; Lv, Q. Hot-Rolled Steel Strip Surface Inspection Based on Transfer Learning Model. *J. Sens.* **2021**, *2021*, 6637252. [CrossRef]
19. Zheng, Y.; Mamledesai, H.; Imam, H.; Ahmad, R. Deep Learning-based Automatic Damage Recognition and Spatial Localization for Remanufacturing/Repair. In Proceedings of the CAD'20, Barelona, Spain, 6–8 July 2020; Volume 18, pp. 381–385. [CrossRef]
20. Zheng, Z.; Qi, H.; Zhuang, L.; Zhang, Z. Automated rail surface crack analytics using deep data-driven models and transfer learning. *Sustain. Cities Soc.* **2021**, *70*, 102898. [CrossRef]
21. Konovalenko, I.; Maruschak, P.; Brezinová, J.; Prentkovskis, O.; Brezina, J. Research of U-Net-Based CNN Architectures for Metal Surface Defect Detection. *Machines* **2022**, *10*, 327. [CrossRef]

22. Litvintseva, A.; Evstafev, O.; Shavetov, S. Real-time Steel Surface Defect Recognition Based on CNN. In Proceedings of the IEEE International Conference on Automation Science and Engineering, Lyon, France, 23–27 August 2021; pp. 1118–1123. [CrossRef]
23. Li, G.; Shao, R.; Wan, H.; Zhou, M.; Li, M. A Model for Surface Defect Detection of Industrial Products Based on Attention Augmentation. *Comput. Intell. Neurosci.* **2022**, *2022*, 9577096. [CrossRef]
24. Pan, Y.; Zhang, L. Dual attention deep learning network for automatic steel surface defect segmentation. *Comput.-Aided Civ. Infrastruct. Eng.* **2022**, *37*, 1468–1487. [CrossRef]
25. Wang, J.; Xu, G.; Yan, F.; Wang, J.; Wang, Z. Defect Transformer: An Efficient Hybrid Transformer Architecture for Surface Defect Detection. *arXiv* **2022**, arXiv:2207.08319. [CrossRef]
26. Gao, L.; Zhang, J.; Yang, C.; Zhou, Y. Cas-VSwin transformer: A variant swin transformer for surface-defect detection. *Comput. Ind.* **2022**, *140*, 103689. [CrossRef]
27. Alzubaidi, L.; Zhang, J.; Humaidi, A.J.; Al-Dujaili, A.; Duan, Y.; Al-Shamma, O.; Santamaría, J.; Fadhel, M.A.; Al-Amidie, M.; Farhan, L. *Review of Deep Learning: Concepts, CNN Architectures, Challenges, Applications, Future Directions*; Springer International Publishing: Berlin/Heidelberg, Germany, 2021; Volume 8. [CrossRef]
28. Tan, C.; Sun, F.; Kong, T.; Zhang, W.; Yang, C.; Liu, C. A survey on deep transfer learning. In Proceedings of the Artificial Neural Networks and Machine Learning–ICANN 2018: 27th International Conference on Artificial Neural Networks, Rhodes, Greece, 4–7 October 2018; Part III 27. Springer International Publishing: Berlin/Heidelberg, Germany, 2018; pp. 270–279. [CrossRef]
29. 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 Proceedings of the Computer Vision—ECCV 2014: 13th European Conference, Zurich, Switzerland, 6–12 September 2014; Part V 13. Springer International Publishing: Berlin/Heidelberg, Germany, 2014; pp. 740–755. [CrossRef]
30. Li, J.; Su, Z.; Geng, J.; Yin, Y. Real-time Detection of Steel Strip Surface Defects Based on Improved YOLO Detection Network. *IFAC-PapersOnLine* **2018**, *51*, 76–81. [CrossRef]
31. Kou, X.; Liu, S.; Cheng, K.; Qian, Y. Development of a YOLO-V3-based model for detecting defects on steel strip surface. *Measurement* **2021**, *182*, 109454. [CrossRef]
32. Xu, Y.; Zhang, K.; Wang, L. Metal surface defect detection using modified yolo. *Algorithms* **2021**, *14*, 257. [CrossRef]
33. Guo, Z.; Wang, C.; Yang, G.; Huang, Z.; Li, G. MSFT-YOLO: Improved YOLOv5 Based on Transformer for Detecting Defects of Steel Surface. *Sensors* **2022**, *22*, 3467. [CrossRef] [PubMed]
34. Saiz, F.A.; Alfaro, G.; Barandiaran, I. An Inspection and Classification System for Automotive Component Remanufacturing Industry Based on Ensemble Learning. *Information* **2021**, *12*, 489. [CrossRef]
35. Bai, J.; Wu, D.; Shelley, T.; Schubel, P.; Twine, D.; Russell, J.; Zeng, X.; Zhang, J. A Comprehensive Survey on Machine Learning Driven Material Defect Detection: Challenges, Solutions, and Future Prospects. *arXiv* **2024**, arXiv:2406.07880.
36. Hütten, N.; Gomes, M.A.; Hölken, F.; Andricevic, K.; Meyes, R.; Meisen, T. Deep Learning for Automated Visual Inspection in Manufacturing and Maintenance: A Survey of Open-Access Papers. *Appl. Syst. Innov.* **2024**, *7*, 11. [CrossRef]
37. Wang, C.-Y.; Bochkovskiy, A.; Liao, H.-Y.M. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In Proceedings of the 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, 17–24 June 2023; pp. 7464–7475. [CrossRef]
38. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Computer Society Conference on Computer Vision. and Pattern Recognition 2016, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788. [CrossRef]
39. Redmon, J.; Farhadi, A. YOLO9000: Better, faster, stronger. In Proceedings of the 30th IEEE Conference on Computer Vision. and Pattern Recognition 2017 (CVPR 2017), Honolulu, HI, USA, 21–26 July 2017; pp. 6517–6525. [CrossRef]
40. Redmon, J. YOLOv3: An Incremental Improvement. *arXiv* **2018**, arXiv:1804.02767.
41. Lhrg, X. YOLOv4: Optimal Speed and Accuracy of Object Detection. *arXiv* **2020**, arXiv:2004.10934v1.
42. Wang, C.-Y.; Liao, H.-Y.M.; Kinsley, H.; Kukieła, D.; Meckel, A. Scaled-YOLOv4: Scaling Cross Stage Partial Network. *arXiv* **2017**, arXiv:2011.08036v2. [CrossRef]
43. Ge, Z.; Liu, S.; Wang, F.; Li, Z.; Sun, J. YOLOX: Exceeding YOLO Series in 2021. *arXiv* **2021**, arXiv:2107.08430. [CrossRef]
44. Wang, C.-Y.; Yeh, I.-H.; Liao, H.-Y.M. You Only Learn One Representation: Unified Network for Multiple Tasks. *arXiv* **2021**, arXiv:2105.04206.
45. Jiang, P.; Ergu, D.; Liu, F.; Cai, Y.; Ma, B. A Review of Yolo Algorithm Developments. *Procedia Comput. Sci.* **2021**, *199*, 1066–1073. [CrossRef]
46. Sari, E.; Belbahri, M.; Nia, V.P. How Does Batch Normalization Help Binary Training? *arXiv* **2019**, arXiv:1909.09139.
47. Severstal. Severstal: Steel Defect Detection. [Dataset]. Available online: https://www.kaggle.com/competitions/severstal-steel-defect-detection/data (accessed on 20 June 2024).
48. Zhang, D.; Song, K.; Xu, J.; He, Y.; Niu, M.; Yan, Y. MCnet: Multiple Context Information Segmentation Network of No-Service Rail Surface Defects. *IEEE Trans. Instrum. Meas.* **2021**, *70*, 5004309. [CrossRef]
49. Božič, J.; Tabernik, D.; Skočaj, D. Mixed supervision for surface-defect detection: From weakly to fully supervised learning. *Comput. Ind.* **2021**, *129*, 103459. [CrossRef]
50. Wieler, F.A.H.M.; Hahn, T. Weakly Supervised Learning for Industrial Optical Inspection [Dataset]. Available online: https://hci.iwr.uni-heidelberg.de/content/weakly-supervised-learning-industrial-optical-inspection (accessed on 20 June 2024).

51. Lv, X.; Duan, F.; Jiang, J.J.; Fu, X.; Gan, L. Deep metallic surface defect detection: The new benchmark and detection network. *Sensors* **2020**, *20*, 1562. [CrossRef]
52. Song, K.; Yan, Y. A noise robust method based on completed local binary patterns for hot-rolled steel strip surface defects. *Appl. Surf. Sci.* **2013**, *285*, 858–864. [CrossRef]
53. Ball Screw Drive Surface Defect Dataset for Classification. Institute for Production Technology. Available online: https://publikationen.bibliothek.kit.edu/1000133819 (accessed on 20 June 2024).
54. Tabernik, D.; Šela, S.; Skvarč, J.; Skočaj, D. Segmentation-based deep-learning approach for surface-defect detection. *J. Intell. Manuf.* **2020**, *31*, 759–776. [CrossRef]
55. Gan, J.; Li, Q.; Wang, J.; Yu, H. A Hierarchical Extractor-Based Visual Rail Surface Inspection System. *IEEE Sens. J.* **2017**, *17*, 7935–7944. [CrossRef]
56. Hu, M.; Ju, X. Two-stage insulator self-explosion defect detection method based on Mask R-CNN. In Proceedings of the 2021 IEEE 2nd International Conference on Intelligent Computing and Human-Computer Interaction (ICHCI), Shenyang, China, 17–19 November 2021; pp. 13–18. [CrossRef]
57. Li, L.; Jiang, Z.; Li, Y. Surface Defect. Detection Algorithm of Aluminum Based on Improved Faster RCNN. In Proceedings of the 2021 IEEE 9th International Conference on Information, Communication and Networks (ICICN), Xi'an, China, 25–28 November 2021; pp. 527–531. [CrossRef]
58. Shorten, C.; Khoshgoftaar, T.M. A survey on Image Data Augmentation for Deep Learning. *J. Big Data* **2019**, *6*, 60. [CrossRef]
59. Inoue, H. Data Augmentation by Pairing Samples for Images Classification. *arXiv* **2018**. [CrossRef]
60. Rožanec, J.M.; Zajec, P.; Theodoropoulos, S.; Koehorst, E.; Fortuna, B.; Mladenic, D. Synthetic Data Augmentation Using GAN For Improved Automated Visual Inspection. *IFAC-Pap.* **2023**, *56*, 11094–11099. [CrossRef]
61. Wang, Y.; Wang, H.; Xin, Z. Efficient Detection Model of Steel Strip Surface Defects Based on YOLO-V7. *IEEE Access* **2022**, *10*, 133936–133944. [CrossRef]
62. Yang, L.; Xu, S.; Fan, J.; Li, E.; Liu, Y. A pixel-level deep segmentation network for automatic defect detection. *Expert. Syst. Appl.* **2023**, *215*, 119388. [CrossRef]
63. Tao, H.; Duan, Q.; Lu, M.; Hu, Z. Learning discriminative feature representation with pixel-level supervision for forest smoke recognition. *Pattern Recognit.* **2023**, *143*, 109761. [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.