







A Novel Deep Learning-based Automatic Damage Detection and Localization Method for Remanufacturing/Repair

Yufan Zheng¹ , Harshavardhan Mamledesai² , Habiba Imam³  and Rafiq Ahmad^{*4} 

¹ Laboratory of Intelligent Manufacturing, Design and Automation (LIMDA), University of Alberta, yufan6@ualberta.ca

² Laboratory of Intelligent Manufacturing, Design and Automation (LIMDA), University of Alberta, mamledes@ualberta.ca

³ Laboratory of Intelligent Manufacturing, Design and Automation (LIMDA), University of Alberta, imam@ualberta.ca

⁴ Laboratory of Intelligent Manufacturing, Design and Automation (LIMDA), University of Alberta, rafiq.ahmad@ualberta.ca

Corresponding author: Rafiq Ahmad, rafiq.ahmad@ualberta.ca

Abstract. Remanufacturing has been considered as an eco-industry, demonstrating environmental and economic benefits. Damage feature inspection is a critical and step in remanufacturing, which establishes the connection between used part and process planning. However, the current inspection method for remanufacturing heavily relies on manual operations. In this study, a deep learning-based damage recognition and spatial localization method is developed. The damage recognition method is based on a Mask-RCNN model to output damage type, 2D damage segments. By mapping the 2D pixel coordinates to the 3D global coordinate system, the spatial coordinate of damage is calculated. With identifying and positioning damages, further automatic repairing/remanufacturing processes can be operated based on these results.

Keywords: Deep learning; Remanufacturing; Repair; Automatic inspection; Mask-RCNN; Spatial localization.

DOI: <https://doi.org/10.14733/cadaps.2021.1359-1372>

1 INTRODUCTION

In contemporary manufacturing, the increasing developments and over-exploitation of resources result in numerous “end-of-life” products. However, the products have not been used thoroughly, and their product life-cycle can be extended by remanufacturing. Remanufacturing has been widely emphasized because it enables the remanufactured product to be sold as a new product and also maintains the intrinsic energy of the “end-of-life” product without creating redundant energy. In the concept of “product life cycle”, remanufacturing extends a product life before its final disposal, as illustrated in Figure 1. The end-of-life components can be re-used, recycled for parts or recycled for materials. It is reported that remanufacturing reduces cost by 50%, energy

by 60%, material by 70% and air pollution by 80% as compared to a conventional manufacturing process [21].

Although significant benefits can be gained from remanufacturing/repair, there are still numerous challenges to implement it in the industry. One of the reasons is that, compared to the manufacturing process, stochastic returns of used parts and their uncontrollable quality condition result in a high degree of uncertainty for the remanufacturing process [16]. The uncertainty surrounding the return of the parts complicates the remanufacturing process. Recently, significant efforts have been devoted to the remanufacturing process plan optimization with uncertainties [25]. These optimization frameworks are initialized with characterized and quantified fault features (e.g. crack, dent, scratch, abrasion). The visual or manual inspection determines the fault feature characterization, which indicates damage type, damage location and damage degree. These three factors play a key role in generating an optimal process plan with different additive operations (e.g. chromium plating, arc welding, cold welding, laser cladding, thermal spraying) and subtractive operations (e.g. milling, grinding) with heuristic algorithms. The current visual or manual inspection methods require extensive human intervention, and the quality of the process is hard to be stable. Therefore, an automated inspection approach for remanufacturing is urgently demanded. For this reason, an increasing level of interest in research on the automated or semi-automated inspection for remanufacturing or repair has been witnessed over recent years [4–6]. By summarizing these research results, to the best of the authors' knowledge, an automatic approach that enables damage recognition and spatial localization simultaneously for remanufacturing has not been discovered. In this study, a deep learning-based damage recognition and spatial localization method is proposed, which can classify different damage features and localize in the global three-dimensional coordinate.

To validate the efficiency of this methodology, this study is implemented in a case study of pipe damage visual inspection for oil industry.

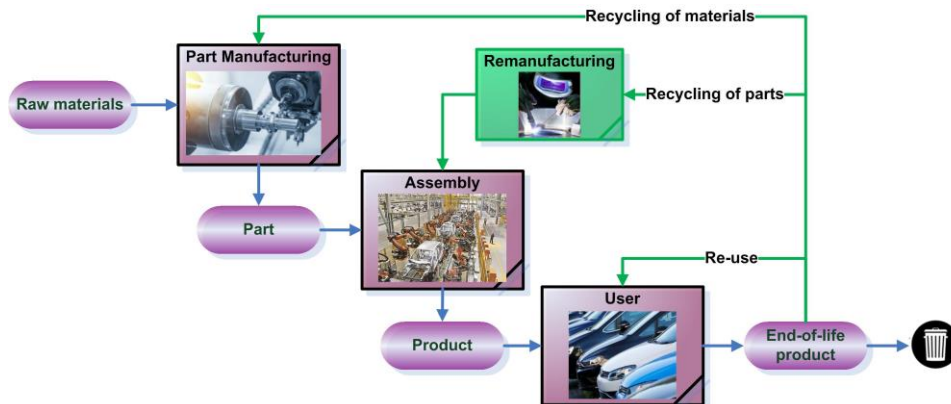


Figure 1: The concept of the product life cycle.

2 LITERATURE REVIEW

There are two categories of damage detections given in recent publication, through collecting point clouds by reverse engineering or images from the damage component. The related works of these two classes of methods are reviewed and summarized as follows.

2.1 Reverse Engineering-based Damage Detection

The reverse engineering techniques enable a quick and accurate acquisition of the three-dimensional (3D) point clouds of the damaged components. Many current studies [28-31] have

introduced reverse engineering techniques in surface modelling to aid the remanufacturing/repair process of damaged parts. Commonly, the process is composed of three steps: 1. data acquisition; 2. comparison of a nominal model and damaged model; 3. repair volume extraction. For data acquisition, a laser triangulation-based or structured light-based scanner device is used to capture the surface geometry of the damaged part in the form of 3D point clouds. The identification and localization of the damaged area are achieved by a registration operation by comparing the nominal CAD model with the model of the damaged part. However, in some cases, the nominal CAD model is not available due to confidentiality issues. A few studies have been focused on CAD free repairing. Wilson et al.[27], Goyal et al. [7], and Piya et al. [13] reconstructed the original turbine blade model by using a prominent cross-section (PCS) method. Li et al. [17] extended the PCS for the reconstruction of other industrial parts, such as a worn gear bracket. Zheng et al. [5] developed a nominal model reconstruction method that can be applied to all primitive-based geometries. After comparing the nominal CAD model and damaged, the repair volume can be extracted by a Boolean operation [17] or distance-based filtering operation [3]. It can be observed that these reverse engineering aided remanufacturing processes are straightforward but complicated and time-consuming. To facilitate the repair process, researchers have been studied on simplifying the damage detection process by segmenting the defective surface directly. Hitchcox and Zhao [12] have developed a quick and accurate surface defect segmentation method from 3D scan data with application to aerospace repair. Especially, the scanning of the entire damaged part can take a couple of hours. Jovančević et al. [14] introduced a novel automatic defect inspection method by analyzing 3D data collected with a 3D scanner. This method is based on estimating the curvature and normal information at every point from the point clouds to identify undesired defects as dents, protrusion or scratches. Borsu et al. [1] extracted the damaged region from input point clouds by estimating the standard deviation of the surface normal vector. 3D point clouds-based damage detection technologies have been widely implemented in other areas such as civil and plant facilities. Kashani & Graettinger [15] introduced a clustering-based feature segmentation method for light detection and ranging (LiDAR) point clouds and applied in detection damages for building roofs. Shinozaki et al. [22] developed an automatic detection method to find scaffolding and wearing on furnace walls from large-scale point clouds. However, there are still some limitations of those methods, such as lacking a generalized algorithm to detect defective regions for all applications, disallowing to classify different classes of damages, performing at low speed due to the computational expense.

2.2 Image-based Damage Detection

Another damage detection method is based on analyzing the input data of images from the damaged components. Deep learning has achieved substantial development in object detection and classification from images in recent years, which uses a series of layers of nonlinear activation functions. With such structure, it enables to integrate feature extraction and classification by optimization, and outputs expected label in the last layer. Benefiting from this effective method, some researchers have been implementing deep learning-based algorithms in defect inspection problems. Masci et al. [20] presented a Max-Pooling Convolutional Neural Network method for classification of 7 different steel defects; however, their work was limited to a shallow neural network. In a modern implementation of a convolutional neural network (CNN) in image classification, Wang et al. [26] presented a CNN-based vision inspection method to identify and classify defective product with high accuracy. However, image classification cannot meet the entire requirement of the task of defect inspection, lacking finding the position of the defect area. Many state-of-the-art object detection methods have been developed using the region-based CNN (R-CNN) architecture. Mask R-CNN as an extension of R-CNN enables simultaneously object detection and instance, segmentation [10]. It has two stages: 1. Images are scanned, and the proposal is generated; 2. The proposal is classified, and the bounding box and mask are generated. Instance segmentation features the potentials to address the localization problems of the defective area in a two-dimensional (2D) aspect. Ferguson et al. [4] introduced an automatic defect detection method to identify casting defects in X-ray images, based on the Mask R-CNN architecture. Zhang et al.

developed a vehicle-damage-detection segmentation algorithm based on transfer learning and an improved Mask RCNN. However, their method can only find the damaged area from the images directly in 2D which has a huge error matching to its position in the real world.

With the development of the techniques of RGB-Depth sensors, the semantic segmentation has a great achievement in indoor scenes [5]. By adding the depth map, the RGB-D image gives the information of the distance of the objects to the camera. In addition, the RGB and depth map have the corresponding relations in pixels. Gupta et al. [8] developed a two-step method to apply different neural networks to RGB and depth map separately to extract the corresponding features separately and classify by support-vector machine (SVM) in the end. Song and Xiao [24] adopt a directional Truncated Signed Distance Function (TSDF) encoding method to train the RGB-D data in the CNN directly and outputs 3D object bounding boxes.

In summary, the exiting image-based damage detection methods only focus on the object detection or semantic segmentation in 2D scenes. However, the localization of the damage area is significant task for repairing and remanufacturing purpose. The development of 3D semantic segmentation with RGB-D image has a great achievement recently. The idea of this study is inspired from these methods. However, most of the 3D semantic segmentation methods are applied for indoor objects. The great performances of these methods are relied on the large RGB-D training data [26]. To the authors' best knowledge, there is no RGB-D database existing for the damage detection problem. It is also difficult to build a large RGB-D dataset for the damage part. Hence, the motivation of this study is developing a novel damage detection approach, which enable the model been trained on a small size RGB data and output the damage class and location.

3 METHODOLOGY

The main objective of this study is to automatically detect damages from a remanufacturing part. The study proposes a detection strategy based on a deep-learning technique to recognize and localize damages. The flowchart is shown in Figure 2. There are four main steps of the process: (1) Data acquisition for the RGB image and depth data by a depth camera; (2) the damage recognition and segmentation using a Mask-RCNN-based method, providing damage segments with recognized damage type; (3) the localization of the damage determined by the integration of damage segments and a point cloud from the depth data. (4) the localization of the damage determined by the integration of damage segments and a point cloud from the depth data.

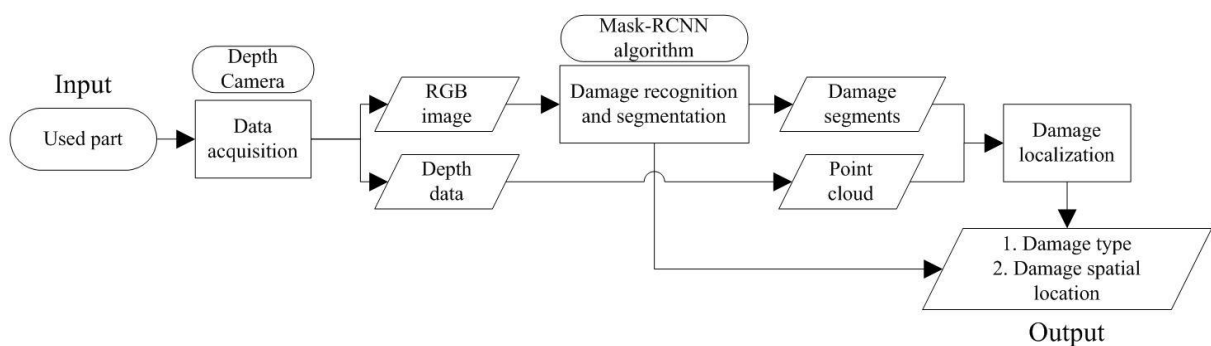


Figure 2: The flowchart of the proposed method.

3.1 Damage Recognition and Classification

In this study, the damage recognition and segmentation method is based on a Mask-RCNN architecture [21]. The proposed damage recognition and segmentation method is illustrated in

Figure 3. As shown, it is composed of four modules: (1) Input the original image to be processed into a pre-trained convolutional backbone to extract features and to obtain a feature map; (2) the region proposal network (RPN) proposes region of interest (ROI) in the feature map with a set of rectangular object proposals; (3) each ROI generates a fixed size feature map by RoIAlign layer; (4) the fixed size feature map goes through two branches of layers for objective classification, frame regression and pixel segmentation.

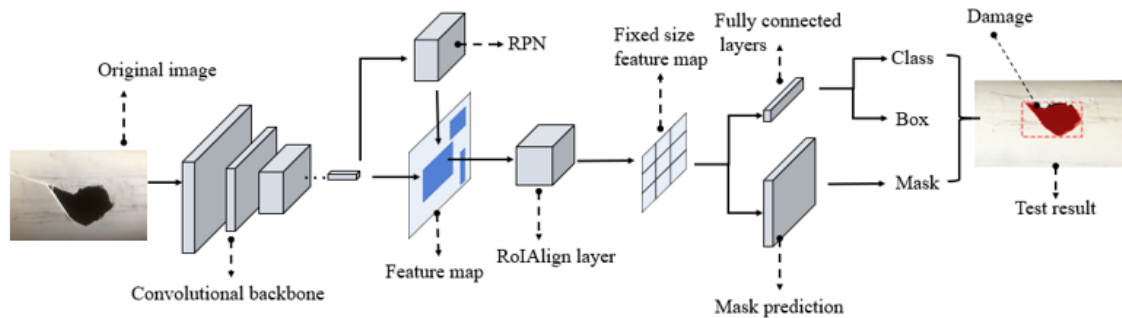


Figure 3: The neural network architecture of the proposed damage recognition and segmentation method.

3.1.1 Convolutional backbone

The convolutional backbone is composed of a series CNN to extract feature maps from the image. The properties of a neural network backbone are characterized by the selection and arrangement of different layers. Deeper networks generally allow to extract more complicate features from the input image, meanwhile stacking more layers will result in issues for training, due to the degradation problem. The residual network (ResNet) was designed to address this problem in deeper neural networks (up to 152 layers) [11] by reformulating its layers as residual learning function with reference to the layer input.

Generally, the Mask RCNN model adopts ResNet101 as the backbone. It is a very deep network with 101 layers and approximately 27 million parameters. In this study, because the damage category is simple and the dataset is limited, a smaller backbone ResNet50 is used to improve the running speed for training. Feature pyramid network (FPN) [18] uses a top-down architecture with lateral connections to build an in-network feature pyramid, which addresses the multi-scale object recognition problem. Overall, this study uses the combination of ResNet50 and FPN as the backbone for feature extraction.

3.1.2 Region Proposal Network

The second module in the proposed damage detection and recognition is RPN. The original image passes through the ResNet50 and FPN convolutional network and outputs a set of convolutional feature maps. In this study, the algorithm uses nine different sizes of anchors as (128*128, 256*256, 512*512) with aspect ratios of (1:1, 1:2, 2:1). Positive or negative anchors are computed by considering the interest-over-union (IoU) between the analyzed anchor and ground-truth bounding boxes on the image. The IoU is calculated by Equation (3.1). In this paper, positive anchors are those that have an IoU is greater or equal to 0.7 in any ground-truth object, and negative anchors are those that have IoU is smaller or equal to 0.3. The anchors with IoU between 0.3 and 0.7 are not considered for the training objective. The positive anchors are then processed to the proposal classification.

$$\text{IoU} = \frac{A_{\text{overlap}}}{A_{\text{union}}} \quad (3.1)$$

where $A_{overlap}$ is the area of overlap and A_{union} is the area of union.

3.1.3 The loss function

The multi-tasking loss function of the Mask R-CNN training process is defined in Equation (3.2), where L is the total training loss; L_{cls} is the classification loss, L_{box} is the bounding-box loss, and L_{mask} is the mask loss.

$$L = L_{cls} + L_{box} + L_{mask} \tag{3.2}$$

The variables for L_{cls} and L_{box} are defined in [6], as shown in Equation (3.3). Each training RoI is labelled with a ground-truth class u and a ground-truth bounding-box regression target v .

$$L_{cls} + L_{box} = L_{cls} p, u + \lambda [u \geq 1] L_{loc} t^u, v \tag{3.3}$$

where u is the label of each training RoI with a ground-truth class; v is a label of each RoI with a ground-truth bounding-box regression target; $t^u = t_x^u, t_y^u, t_w^u, t_h^u$ specifies a scale-invariant translation and log-space height/width shift relative to u class; $p = p_0, \dots, p_K$ represents the probability distribution over $K + 1$ categories; $[u \geq 1]$ denotes the Iverson bracket indicator function that evaluates to 1 when $u \geq 1$ and 0 otherwise.

The bounding-box regression $L_{loc}(t^u, v)$ is shown in:

$$L_{loc}(t^u, v) = \sum_{i \in \{x, y, w, h\}} \text{smooth}_{L_i}(t_i^u - v_i) \tag{3.4}$$

where:

$$\text{smooth}_{L_i}(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise} \end{cases} \tag{3.5}$$

The L_{mask} is calculated by taking the average cross-entropy of all pixels on the RoI, as:

$$L_{mask} = -\frac{1}{N} [y_i \ln a_i + 1 - y_i \ln 1 - a_i] \tag{3.6}$$

$$y_i = 1 / (1 + e^{-x_i})$$

$$a_i = 1 / (1 + e^{-b_i})$$

where x_i and b_i are the prediction value and true value of the i -th pixel in the positive RoI, respectively, N indicates the number of pixels in the positive RoI.

3.2 Spatial Localization

Spatial localization of the damaged area is achieved by finding the mapping relations between the 2D coordinates in the image and 3D spatial coordinates by the depth sensor model, as shown in Equation (3.7).

$$z_c \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{f_x}{dx} & 0 & u_0 \\ 0 & \frac{f_y}{dy} & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} R_z R_y R_x & T \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{3.7}$$

$$R_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta_x & -\sin \theta_x \\ 0 & \sin \theta_x & \cos \theta_x \end{bmatrix}; R_y = \begin{bmatrix} \cos \theta_y & 0 & \sin \theta_y \\ 0 & 1 & 0 \\ -\sin \theta_y & 0 & \cos \theta_y \end{bmatrix}; R_z = \begin{bmatrix} \cos \theta_z & -\sin \theta_z & 0 \\ \sin \theta_z & \cos \theta_z & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$T = [t_x \ t_y \ t_z]^T$$

where u and v are the 2D image coordinates; u_0 and v_0 are the origin of the 2D coordinate system; f_x and f_y are the focal length along x and y direction, respectively; $R_z R_y R_x$ and T are the rotation matrix and translation matrix from the camera coordinate system to the global coordinate system, X, Y, Z are the 3D coordinates under global coordinate. M and m represent the location of the pixel in 3D global coordinate and image, respectively. z_c is the distance of the image to the camera. The illustration is shown in Figure 4.

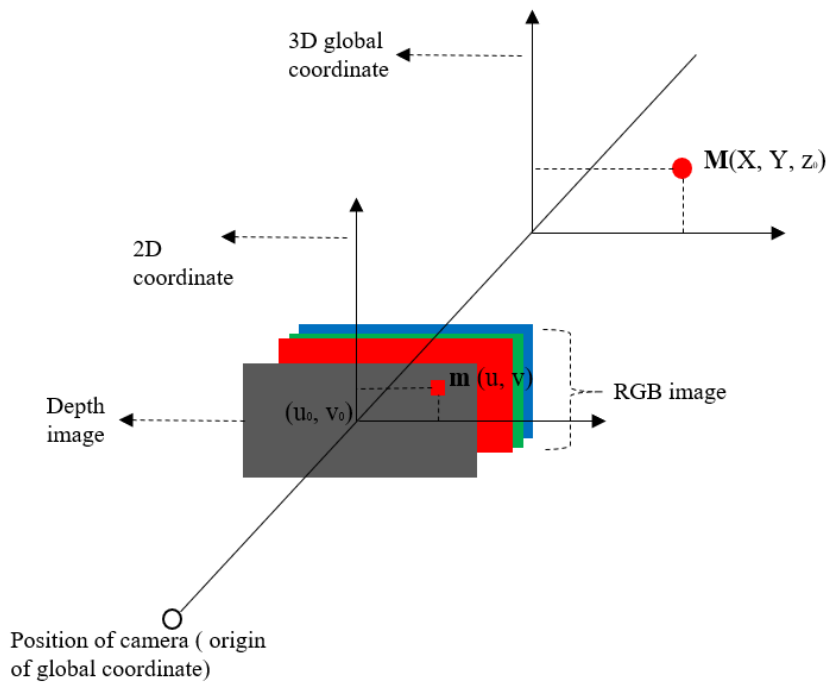


Figure 4: An illustration of the mapping of depth and RGB image coordinate to xyz coordinate.

To simplify this problem, the authors coincide the camera coordinate system and the global coordinate system and Equation (3.7) can be derived as:

$$z_c \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{f_x}{dx} & 0 & u_0 \\ 0 & \frac{f_y}{dy} & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ z_c \\ 1 \end{bmatrix} \quad (3.8)$$

Then, the 3D coordinates of the damaged area can be calculated as:

$$X = \frac{u - u_0 \cdot z_c \cdot dx}{f_x}; \quad Y = \frac{v - v_0 \cdot z_c \cdot dy}{f_y}; \quad Z = z_c \quad (3.9)$$

4 EXPERIMENTAL RESULTS AND ANALYSIS

4.1 Transfer Learning

Deep learning requires a large number of input images as training data, but for some applications, it is very difficult to find enough images. Transfer learning provides an alternative strategy to address this problem. It is possible to reuse a pre-trained CNN weight as a starting point for another training task, instead of building a CNN from scratch. In this study, the training model was initialized using the weights from a ResNet-101 network, which was trained on the COCO dataset [19]. COCO dataset has 330K images with 1.5 million object instances for 80 object categories. Therefore, a good performance pre-trained model can be obtained from this dataset. Using migration learning from the pre-trained model can increase the efficiency of training significantly than starting from scratch.

4.2 Dataset Building

The images with damaged pipes were collected by a GigE DFK 33GD006 image sensor with TCL 3520 5MP lens with a 35 mm focal length, and the setup is shown in Figure 5. The entire dataset includes training, validation and testing datasets with the resolution of 1920*1080 images. The dataset is collected from 30 damaged pipes and each pipe has 3 portions of damage with different sizes. The experiment collected 220 images (160 for the training dataset, 40 for validation dataset and 20 for testing dataset). The training and validation images were annotated according to their damaged areas by polygon shapes using the free annotation software VGG Image Annotator [2]. It labelled the images with the JSON file which contains a class of damage and damage region. Figure 6 gives examples of annotated images.



Figure 5: Dataset acquisition setup.

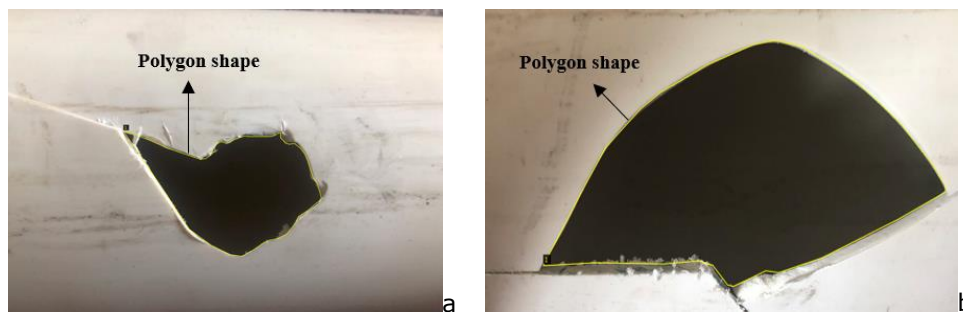


Figure 6: Annotation of damaged areas by polygon shapes.

4.3 Experimental Environment

The experiments were conducted using Mask-RCNN, Matlab2019a, CUDA 10.0, TensorFlow 1.14.0, CuDNN 6.5 on a desktop computer equipped with an Intel Core i5-8600K 3.60 GHz CPU, 16 GB DDR4 ram memory, Nvidia GTX 1060 with 6 GB video ram GPU, under an operating system of Ubuntu 16.04 64 bit. The pre-defined parameters for the damage detection and classification model are shown in Table 1.

<i>Parameter</i>	<i>Value</i>
Batch size	30
Learning rate	0.01
Learning Momentum	0.9
Mask pool size	14
Pool size	7
Step per epoch	200
Detection minimum confidence	0.9
Number of classes	2
epoch	30

Table 1: The pre-defined parameters for damage detection and classification.

In this study, Microsoft Kinect V1 was used as the depth camera for testing. The technical specification of it is presented in Table 2. It outputted RGB image (640*840*3) and depth image (640*840), as shown in Figure 7.

<i>Kinect V1</i>	<i>Specifications</i>
Max. resolution of the colour sensor	1280*960
Max. resolution of the depth sensor	640*480
Viewing angle	43° vertical x 57° horizontal
Vertical tilt range	±27°
Frame rate	30 frames per second (FPS)

Table 2: The pre-defined parameters for damage detection and classification.

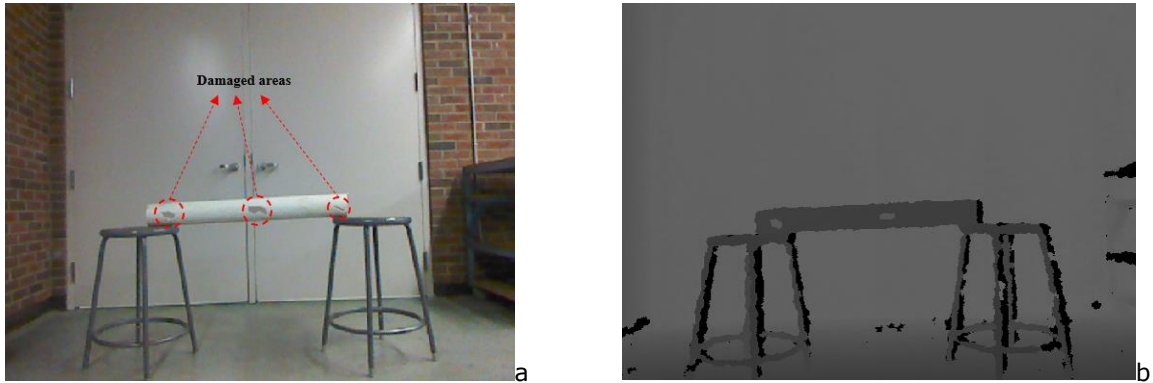


Figure 7: RGB image (a) and depth image (b).

By implementation of the Equation (3.7)-(3.9), the point cloud data (defined as *pointcloud*) was calculated from the RGB image and depth image, as shown in Figure 8. The data structure of *pointcloud* includes *Location* ($480 \times 640 \times 3$), *Color* ($480 \times 640 \times 3$), *Count* (positive integer), *XLimits* (1×2), *YLimits* (1×2), *ZLimits* (1×2). In the data of *Location*, each entry specifies the x , y , and z coordinates of a point in the 3D coordinate space. Therefore, each pixel in the RGB image can be mapped to the *pointcloud.Location* to find their x , y , z coordinates in the 3D space.

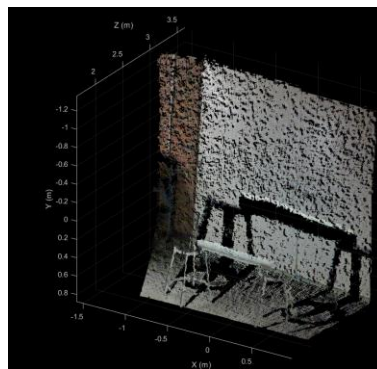


Figure 8: Point cloud dataset.

4.4 Results and Analysis

For the damaged area detection and classification algorithm, after 30 epochs of training, the convergence history of the model loss for both training and validation samples are plotted in Figure 9. It can be observed that the loss for training and validation achieved 0.1612 and 0.5744, respectively. The accuracies in this study were in segment-wise evolution. The accuracies had achieved 99.57% and 87.61% for training and validation datasets. Considering the size of the training dataset, the validation accuracy is acceptable. y would be improved. The authors also had tried changed hyperparameters such as batch size, learning rate, and activation function to improve the performance of the model but achieved limited benefit. Therefore, in this study, increasing the size of the training dataset would be the most effective method to improve the accuracy further. Some examples from the damaged area detection algorithm are shown in Figure 10.

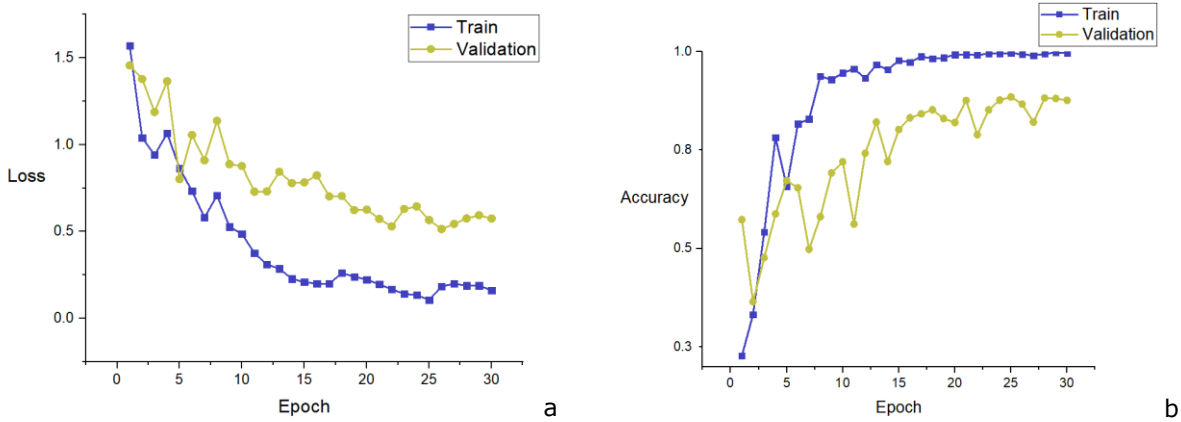


Figure 9: Convergence histories for loss (a) and accuracy (b) after 30 epochs.

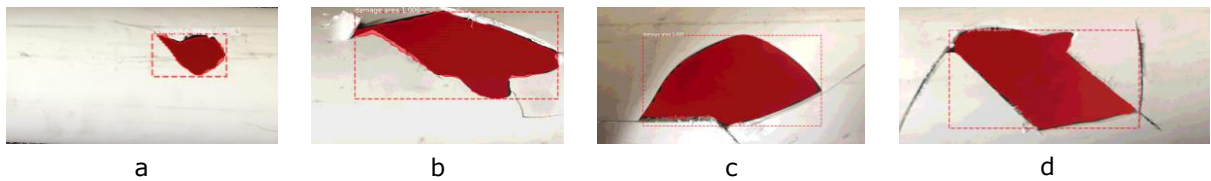


Figure 10: Example detections of the damaged area from the pipes.

The authors tested the performance of the 3D localization in the proposed method by calculating the centroid position of the damaged area. The localization error is defined as follows:

$$\Delta = \sqrt{(x_i - x)^2 + (y_i - y)^2 + (z_i - z)^2} \tag{4.1}$$

where x_i, y_i, z_i are the coordinate of the estimated centroid position and x, y, z are the coordinate of centroid position from manual measurements. The average relative error is defined as:

$$\sigma = \frac{\sum_{i=1}^n \Delta_i}{n\sqrt{x^2 + y^2 + z^2}} \tag{4.2}$$

By conducting measurements for five samples, the manual measurement of the centroid point, estimation of the centroid position by the proposed method were recorded. For each sample, the estimation of the centroid position was calculated by ten times. The results are presented in Table 3. From [7], the proposed had achieved higher error than the traditional damage localization method (around 5 mm). However, the traditional damage localization approach costs a few hours in scanning and around 2000 s for registration. Therefore, the proposed method represents a much higher efficiency than the traditional method. The resolution of the depth sensor impacts the accuracy of the results strongly. High-accuracy 3D depth sensor (such as 2540*1600) can be easily adapted in this study to acquire a lower error of the damage area localization, which will be revealed in the future work.

#	Manual	Average estimation	Average speed	localization	average
---	--------	--------------------	---------------	--------------	---------

	measurement (cm)	by our method (cm)	by our method(s)	error (cm)	relative error (%)
1	(8.2, 2.2, 20.1)	(8.8, 2.4, 20.8)	1.45	0.943	4.322
2	(20.6, 6.8, 20.7)	(19.2, 6.0, 22.2)	1.48	2.202	7.344
3	(10.2, 4.2, 23.2)	(12.2, 4.8, 25.2)	1.42	2.891	11.253
4	(16.3, 2.3, 33.2)	(16.5, 2.3, 35.2)	1.48	2.010	5.424
5	(10.4, 20.2, 18.8)	(11.4, 21.8, 19.8)	1.47	2.135	7.244

Table 3: Results of experiments for the 3D localization.

5 CONCLUSION

Remanufacturing/repairing has been considered a green manufacturing strategy since it reduces cost, energy, material consumption and air pollution significantly compared to traditional manufacturing. Damage detection is the primary step in remanufacturing to make the decision of a remanufacturing strategy. However, the current damage detection method relies heavily on manual operations which is time-consuming. The motivation of this study is developing a novel damage detection approach, which performs damage classification and 3D localization simultaneously.

In order to address these problems, this paper proposes an efficient deep learning-based damage detection and localization method. In the first step, the RGB image and depth image are acquired by a depth camera. Then, training data and validation data are collected to train the Mask-RCNN-based model to obtain optimized weight. The RGB image acquired processed in the damage recognition and segmentation algorithm, providing damage segments with recognized damage types. In the last, the 3D position of the damage is determined by the integration of damage segments and a point cloud from the depth data.

The accuracy of the damage detection can be improved by increasing the training data size. And, the error of damage localization can be reduced by implementing a high-accuracy depth sensor, which will be the focus of future work.

The current remanufacturing/repair industry relies on visual inspection to determine the damage type, damage location and damage degree to schedule the process plans. The study has potentials to perform damage detection to output the damage type, and location. In the future work, a systematic method to determine the damage degree can be investigated.

ACKNOWLEDGMENTS

We express our appreciation to the other team members in the Laboratory of Intelligent Manufacturing, Design and Automation (LIMDA) group for sharing their wisdom during the research. The authors would like to acknowledge NSERC (Grant No. RGPIN-2017-04516 and CRDPJ 537378-18) for funding this project.

Yufan Zheng, <https://orcid.org/0000-0002-3561-3734>

Harshavardhan Mamedesai, <https://orcid.org/0000-0003-2198-9519>

Habiba Imam, <https://orcid.org/0000-0001-5152-948X>

Rafiq Ahmad, <https://orcid.org/0000-0001-9353-3380>

REFERENCES

- [1] Borsu, V.; Yogeswaran, A.; Payeur, P.: Automated surface deformations detection and marking on automotive body panels, IEEE International Conference on Automation Science and Engineering, 2010, 551–6. <https://doi.org/10.1109/COASE.2010.5584643>

- [2] Dutta, A.; Zisserman, A.: The VIA annotation software for images, audio and video, Proceedings of the 27th ACM International Conference on Multimedia, 2019, 2276–9. <https://doi.org/10.1145/3343031.3350535>
- [3] Feng, C.; Liang, J.; Gong, C.; Pai, W.; Liu, S.: Repair volume extraction method for damaged parts in remanufacturing repair, International Journal of Advanced Manufacturing Technology, 98, 2018, 1523–36. <https://doi.org/10.1007/s00170-018-2300-7>
- [4] Ferguson, M.; A, R.; Lee, Y.; Law K.: Detection and segmentation of manufacturing defects with convolutional neural networks and transfer learning. Smart and Sustainable Manufacturing Systems, 1, 2018. <https://doi.org/10.1520/ssms20180033>
- [5] Fooladgar, F.; Kasaei, S.: A survey on indoor RGB-D semantic segmentation: from hand-crafted features to deep convolutional neural networks, Multimedia Tools and Applications, 79, 2020, 4499–524. <https://doi.org/10.1007/s11042-019-7684-3>
- [6] Girshick, R.: Fast R-CNN, Proceedings of the IEEE International Conference on Computer Vision, 2015, 1440–1448. <https://doi.org/10.1109/ICCV.2015.169>
- [7] Goyal, M.; Murugappan, S.; Piya, C.; Benjamin, W.; Fang, Y.; Liu, M.: Towards locally and globally shape-aware reverse 3D modeling, Computer Aided Design, 44, 2012, 537–53. <https://doi.org/10.1016/j.cad.2011.12.004>
- [8] Gupta, S.; Girshick, R.; Arbeláez, P.; Malik, J.: Learning rich features from RGB-D images for object detection and segmentation, European Conference on Computer Vision, 2014, 345–360. https://doi.org/10.1007/978-3-319-10584-0_23
- [9] Hascoët, J.; Touzé, S.; Rauch, M.: Automated identification of defect geometry for metallic part repair by an additive manufacturing process, Welding in the World, 62(2), 2018, 229–241. <https://doi.org/10.1007/s40194-017-0523-0>
- [10] He, K.; Gkioxari, G.; Dollar, P.; Girshick R.: Mask R-CNN, Proceedings of the IEEE International Conference on Computer Vision, 2017, 2980–8. <https://doi.org/10.1109/ICCV.2017.322>
- [11] He, K.; Zhang, X.; Ren, S.; Sun, J.: Deep residual learning for image recognition, Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition; 2016, 770–778. <https://doi.org/10.1109/CVPR.2016.90>
- [12] Hitchcox, T.; Zhao, Y.F.: Random walks for unorganized point cloud segmentation with application to aerospace repair, Procedia Manufacturing, 26, 2018, 1483–91. <https://doi.org/10.1016/j.promfg.2018.07.093>
- [13] Jiang, Z.; Zhou, T.; Zhang, H.; Wang, Y.; Cao, H.; Tian, G.: Reliability and cost optimization for remanufacturing process planning, Journal of Cleaner Production, 135, 2016, 1602–1610. <http://doi.org/10.1016/j.jclepro.2015.11.037>
- [14] Jovančević, I.; Pham, H.H.; Orteu, J.J.; Gilblas, R.; Harvent J.; Maurice X.: 3D point cloud analysis for detection and characterization of defects on airplane exterior surface, Journal of Nondestructive Evaluation, 36, 2017. <https://doi.org/10.1007/s10921-017-0453-1>
- [15] Kashani, A.; Graettinger A.: Cluster-based roof covering damage detection in ground-based lidar data. Automation in construction, 58, 2015, 19–27. <https://doi.org/10.1016/j.autcon.2015.07.007>
- [16] Lee, C.; Woo, W.; Roh, Y.: Remanufacturing: trends and issues. International Journal of Precision Engineering and Manufacturing - Green Technology, 4(1), 2017, 113–125. <http://doi.org/10.1007/s40684-017-0015-0>
- [17] Li, L.; Li, C.; Tang, Y.; Du, Y.: An integrated approach of reverse engineering aided remanufacturing process for worn components, Robotics and Computer-Integrated Manufacturing, 48, 2017, 39–50. <https://doi.org/10.1016/j.rcim.2017.02.004>
- [18] Lin, T.; Dollár, P.; Girshick, R.; He, K.; Hariharan B.; Belongie S.: Feature pyramid networks for object detection, Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, 936–944. <https://doi.org/10.1109/CVPR.2017.106>
- [19] Lin, T.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan D.: Microsoft COCO: common objects in context, European Conference on Computer Vision, 2014, 740–55. https://doi.org/10.1007/978-3-319-10602-1_48

- [20] Masci, J.; Meier, U.; Ciresan, D.; Schmidhuber, J.; Fricout, G.: Steel defect classification with Max-Pooling Convolutional Neural Networks, Proceedings of the International Joint Conference on Neural Networks, 2012, 10–5. <https://doi.org/10.1109/IJCNN.2012.6252468>
- [21] Piya, C.; Wilson, J.; Murugappan, S.; Shin, Y.; Ramani, K.: Virtual repair: Geometric reconstruction for remanufacturing gas turbine blades. ASME 2011 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, , 9, 2011, 895–904. <https://doi.org/10.1115/DETC2011-48652>
- [22] Shinozaki, Y.; Kohira, K.; Masuda, H.: Detection of deterioration of furnace walls using large-scale point-clouds. Computer Aided Design and Application, 15, 2018, 575–84. <https://doi.org/10.1080/16864360.2017.1419645>
- [23] Song S.; Lichtenberg S.; Xiao J.: SUN RGB-D: A RGB-D scene understanding benchmark suite, The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, 567–76. <https://doi.org/10.1109/CVPR.2015.7298655>
- [24] Song, S.; Xiao J.: Deep sliding shapes for amodal 3D object detection in RGB-D images. The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, 808–16. <https://doi.org/10.1109/CVPR.2016.94>
- [25] Wang, H.; Jiang, Z.; Zhang, X.; Wang, Y.; Wang, Y.: A fault feature characterization-based method for remanufacturing process planning optimization, Journal of Cleaner Production, 161, 2017; 708–719. <http://doi:10.1016/j.jclepro.2017.05.178>
- [26] Wang, J.; Fu, P.; Gao, R.: Machine vision intelligence for product defect inspection based on deep learning and Hough transform, Journal of Manufacturing Systems 51, 2019, 52–60. <https://doi.org/10.1016/j.jmsy.2019.03.002>
- [27] Wilson J.; Piya C.; Shin Y.; Zhao F.; Ramani K.: Remanufacturing of turbine blades by laser direct deposition with its energy and environmental impact analysis. Journal of Cleaner Production, 80, 2014, 170–8. <https://doi.org/10.1016/j.jclepro.2014.05.084>
- [28] Zhang, X.; Li, W.; Chen, X.; Cui, W.; Liou, F.: Evaluation of component repair using direct metal deposition from scanned data, International Journal of Advanced Manufacturing Technology, 95, 2017, 1–14. <https://doi.org/10.1007/s00170-017-1455-y>
- [29] Zhang, X.; Li, W.; Liou, F.: Damage detection and reconstruction algorithm in repairing compressor blade by direct metal deposition, The International Journal of Advanced Manufacturing Technology, 95, 2017, 2393-2404. <http://doi:10.1007/s00170-017-1413-8>
- [30] Zheng, Y.; Liu, J.; Liu, Z.; Wang, T.; Ahmad R.: A primitive-based 3D reconstruction method for remanufacturing, The International Journal of Advanced Manufacturing Technology, 103, 2019, 3667–3681. <http://doi:10.1007/s00170-019-03824-w>
- [31] Zheng, Y.; Qureshi, A.J.; Ahmad, R.: Algorithm for remanufacturing of damaged parts with hybrid 3D printing and machining process, Manufacturing Letters, 15, 2018, 38–41. <https://doi.org/10.1016/j.mfglet.2018.02.010>