IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 17 March 2024, accepted 23 May 2024, date of publication 30 May 2024, date of current version 6 June 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3406989

APPLIED RESEARCH

Method and Application of Spraying Developer Defect Detection in Three-Dimensional Optical Scanning Measurement Process Based on YOLO v7

ZHANHUI WANG[1,2], QISHENG ZHAO[1], AND CHAOJIE FENG[1]
[1]School of Mechanical Engineering, Tianjin Sino-German University of Applied Sciences, Tianjin 300350, China
[2]School of Materials Science and Engineering, Yanshan University, Qinhuangdao 066000, China

Corresponding author: Zhanhui Wang (wangzhanhui@tsguas.edu.cn)

This work was supported in part by Tianjin Education Commission Research Program Project 2018KJ262, in part by the Science & Technology Development Fund of Tianjin Education Commission for Higher Education under Grant 2021KJ099, and in part by Tianjin Science and Technology Planed Project 23YDTPJC00940.

ABSTRACT In industrial production, three-dimensional scanning and inspection technology is widely applied, with spraying developer playing a critical role in this process. During the application of spraying developer, the spraying defects generated on the surface of workpieces can greatly affect the accuracy and integrity of the three-dimensional scanning data construction for the products. Currently, the detection of these spraying defects heavily relies on manual visual observation, a process that is both time-consuming and labor-intensive. Moreover, there is no standardized approach, leading to uncertainty in the subsequent construction of three-dimensional scanning data. To address this issue, this study proposes a deep learning algorithm based on YOLO v7 target detection. The research focuses on commonly used inspection parts in the field of three-dimensional optical scanning measurement, constructing a dataset specifically for spraying defects. Real-time target detection is then conducted on the workpieces within an actual production environment. Through model evaluation, it was found that the neural network in this study achieved a remarkable accuracy of 0.996 on the test set, with the highest confidence level of 0.96 during real-time inspection of the workpieces. These results demonstrate the superior accuracy of the proposed method in detecting defect targets, thereby offering valuable insights for evaluating spraying defects and enhancing the quality of three-dimensional optical scanning data construction.

INDEX TERMS Three-dimensional scanning, spraying developer defects, deep learning, computer vision, image detection.

I. INTRODUCTION

Three-dimensional (3D) optical scanning has emerged as a prominent measurement technique in various fields, experiencing rapid development over the past decade [1], [2], [3], [4], [5], [6]. Compared to touch probe CMM systems, 3D optical scanning systems offer advantages of high efficiency and quality, particularly in accurately measuring complex shapes [7], [8], [9]. These systems find applications in diverse industries such as automotive, medical, architecture, and historic preservation. In the automotive industry, for instance, 3D optical scanners are employed to capture the intricate shape of objects, facilitating the modeling process. They generate high-resolution models by capturing millions of points within seconds, resulting in point cloud data. This data can then be transferred to a CAD system for tasks such as 3D surface or solid modeling, finite element analysis, tool design, and tool path generation [10], [11], [12]. Furthermore, the triangulation-based principle underlying 3D optical scanning measurements offers excellent scalability.

The associate editor coordinating the review of this manuscript and approving it for publication was Sudhakar Radhakrishnan.

This allows these systems to be utilized across a wide range of scales, making them a valuable complement to traditional contact measurement equipment [13].

In industrial production, three-dimensional optical scanning and measurement systems are widely employed to acquire point cloud data by projecting lasers onto objects and processing the reflected light. These systems have matured significantly in terms of optical control technology, offering advantages over contact-based systems. They exhibit reduced sensitivity to operator influence [12], [14], [15]. However, several factors can still impact the quality of point cloud data, and the presence of defects in the data can adversely affect subsequent data processing [7], [16], [17], [18].

These factors can be attributed to both internal and external influences on the scanning process. Internal factors include scanner resolution, accuracy, and other equipment-related parameters. External factors encompass parameters like the selection of appropriate scanning settings, environmental lighting conditions, surface characteristics of the scanned objects (such as color, glossiness, roughness, and shape), and the relative positioning of the sensor in relation to the surface [9], [19], [20]. Investigating these factors and accurately configuring them is of significant importance in industrial applications.

Among the external factors, controlling the surface characteristics of the scanned objects presents a particular challenge. The surface properties determine the reflection behavior after laser illumination, which has a substantial impact on the quality of the acquired data [7], [9], [20]. To achieve high-quality reflected light, researchers have explored methods such as utilizing diffuse reflection from the surface or employing specific filters on CCD sensors [20], [21]. However, for practical purposes and ease of implementation, obtaining good diffuse reflection characteristics on the surface is preferable within a given scanning setup. For smooth surfaces, especially metal surfaces, applying spraying developer before measurement creates a white matte layer, facilitating diffuse reflection [12]. The thickness of the developer typically ranges from 5 to 18 μm, with an average value of 11 μm [22]. In such cases, the optical system is calibrated based on factors such as laser power and exposure time to acquire accurate point cloud data. However, it is crucial to note that defects generated on the surface during the spray developer application process can significantly affect the accuracy and integrity of the three-dimensional scanning data construction for the product [6], [7], [9]. Section II-A2 of this paper will provide a detailed explanation of the impact mechanism of spray defects on data construction.

Currently, the detection of spray developer defects relies mainly on manual visual observation, and the quality of the point cloud data obtained from three-dimensional optical scanning is dependent on the skills and expertise of the spray application operator [7], [12], [19]. The process of detecting spray developer defects is time-consuming, labor-intensive, and lacks a standardized approach, leading to uncertainties in the subsequent three-dimensional scanning measurement results.

Deep learning [23] is an exciting field within machine learning (ML) that encompasses artificial neural networks with multiple hidden layers. It has found extensive applications in target detection [24]. In recent years, the computational power of computers has significantly increased, thanks to the era of big data and the rapid development of computer graphics cards. This has accelerated the progress of artificial intelligence research, with computer vision being widely adopted in various industrial inspection scenarios. For instance, R-CNN, initially introduced by Girshick et al. [25], pioneered the two-stage detection approach. Jiang et al. [26] achieved impressive results by applying Faster R-CNN [27] to facial detection. However, the inclusion of a large number of predefined anchor points in Faster R-CNN resulted in increased computational complexity. Moreover, while the use of deep convolutional networks contributed to superior detection accuracy, the accompanying redundant operations significantly escalated space and time costs, making it challenging to deploy this method in real-world industrial scenarios. Law et al. [28] proposed a one-stage object detection method called CornerNet, along with a novel pooling technique called Corner Pooling. Nonetheless, methods based on keypoints often encounter a considerable number of incorrect bounding boxes, limiting their performance and failing to meet the high-performance requirements of defect detection models utilizing spray developer. Building upon CornerNet, Duan et al. [29] developed the CenterNet framework to enhance accuracy and recall. They designed two custom modules that exhibit stronger robustness to feature-level noise. However, Anchor-Free methods involve a process of combining the first two keypoints, and their simplicity in network structure, time-consuming operations, low speed, and unstable measurement results hinder their ability to meet the high-performance and high-accuracy requirements of real-time object detection in manufacturing.

Popular single-stage object detection algorithms, such as the YOLO series and SSD, outperform multi-stage methods [30], [31], [32]. These algorithms only require feature extraction once to accomplish object detection. Qiu and Lau [33] compared multiple algorithms and discovered that YOLO v2 and YOLO v4-tiny, based on ResNet50, achieved exceptional accuracy in detecting small cracks. Li et al. [34] improved the feature fusion module of the YOLOX object detection algorithm, exhibiting significant advantages over other detection algorithms in identifying four types of defects in rubber wood. Carrasco et al. [35] proposed a lightweight deep object detection model, based on YOLO-v5, for vehicle detection in parking lots, including large, small, and mini cars, resulting in a 33% performance improvement compared to other algorithms. Wang et al. [36] found that YOLO v7 achieved an average precision of 95.5% and an FPS of 54 in weld defect detection research. Zheng [37] achieved an average precision of 93.8% in insulation detection using an

improved version of YOLO v7. Dehaerne et al. [38] found that YOLO v7 struck a good balance between accuracy and inference time in semiconductor defect detection research.

In conclusion, spraying developer is a necessary component in the process of three-dimensional optical scanning. The defects generated during the spraying process affect the surface characteristics of objects, which in turn impact the effectiveness of collecting three-dimensional optical scan data. Manual detection is predominantly used for identifying spray defects, but it suffers from issues such as low efficiency, lack of standardized evaluation criteria, and unstable detection accuracy. Therefore, accurately and rapidly identifying and repairing spraying developer defects before data acquisition is crucial for obtaining high-quality three-dimensional scan data. Single-stage object detection algorithms exhibit high performance and accuracy in object detection. In this study, considering the process of spraying defects generation and the impact of spray defects on the construction of three-dimensional optical scan data, we propose an improved inference mechanism-based YOLO v7 algorithm [39]. We create a dataset of spraying developer defects and perform real-time object detection on workpieces in actual production. Through model evaluation, we obtain a technical solution for accurately identifying spraying defects, thereby enhancing the accuracy and integrity of three-dimensional optical scan data. This has significant value in practical engineering applications.

II. EXPERIMENTAL METHODS

A. CONSTRUCTION OF 3D OPTICAL SCAN DATA

According to the introduction provided in this paper, previous studies have extensively discussed the use of spray developers to enhance the reflective properties and surface texture of target objects, thereby improving the accuracy of 3D optical scanning measurements. However, there is a lack of research addressing the impact of spray developer defects on the construction of 3D optical scanning measurement data. Therefore, it is crucial to elucidate the steps involved in the spray developer process and the mechanism by which spray developer defects influence the construction of accurate 3D optical scanning measurement data in this paper.

1) DEVELOPER SPRAYING PROCEDURE AND POSSIBLE DEFECTS

The process of spraying developer in the 3D optical scanning measurement system involves applying a layer of developer onto the surface of the object being measured prior to the actual measurement. The developer used is typically a white powder which is sprayed onto the surface of the test object. This helps to achieve a uniform and diffuse reflection on the surface, enhancing the quality of the scanning data by minimizing color variations, reflections, and other imperfections that may occur. As a result, it becomes easier to scan objects with black, reflective, or transparent surfaces, allowing for the acquisition of high-quality point cloud data. This ultimately improves the precision of measurement by enhancing the scanning system's ability to accurately identify surface features on the workpieces [22]. The parts of the object after the developer has been sprayed are illustrated in Fig. 1.

FIGURE 1. Part after spraying developer.

The process of spraying developer typically involves the following steps:

Preparation: Begin by thoroughly cleaning the surface of the workpiece to ensure it is free from any oil or dust particles. This step is crucial to ensure proper adhesion of the developer.

Setting up the developer spraying equipment: Load the white powder into the powder spraying equipment. Alternatively, you may opt to purchase spraying equipment that already has pressure in the bottle. Adjust the angle and pressure of the spraying equipment according to the shape and size of the workpiece.

Performing the developer spraying operation: Align the developer spraying equipment with the surface of the workpiece and commence the spraying process. It is important to maintain a uniform thickness of developer throughout the spraying operation, ensuring that the entire surface of the workpiece is adequately covered.

Post-treatment of the sprayed developer: Once the developer spraying is complete, carefully inspect the quality of the sprayed developer. Any areas that require corrective measures or additional powder coverage should be addressed promptly to achieve a uniform coating across the entire surface of the workpiece.

During the process of spraying developer, several defects may occur, including the following:

Uneven developer spraying: This can result in areas where the powder layer is too thick or too thin.

Missed areas in developer coverage: Some regions of the workpiece may be left uncovered during the spraying process.

Rough powder particles: The powder material may clump together, forming rough patches with small particles, resulting in a rough and speckled appearance.

Powder shedding: Insufficient curing or poor bonding can cause the powder to detach from the surface, leading to powder shedding.

Surface abrasions and scratches: The sprayed developer parts may exhibit scratches, abrasions, or other forms of surface damage.

Among the mentioned defects, powder shedding, surface abrasions, and scratches have a more significant impact on the accuracy of the test results and will be thoroughly discussed in section II-A2 of this paper.

To prevent defects during the process of spraying developer, it is crucial for the operator to possess a certain level of skill and experience in handling the spraying equipment and adhere strictly to the operating specifications. Additionally, a meticulous manual inspection of the parts is required after the developer has been sprayed, demanding a high level of technical expertise. In this experiment, a device equipped with high-pressure gas inside the bottle is employed to achieve the developer spraying on the parts, as depicted in Fig. 2.

FIGURE 2. Developer spraying operation on parts using a device with its own high-pressure gas in a bottle.

2) MECHANISMS OF THE INFLUENCE OF SPRAY DEVELOPER DEFECTS ON THE CONSTRUCTION OF 3D OPTICAL SCAN DATA

Three-dimensional optical scanning measurement systems provide a contactless, rapid, and highly precise means of obtaining measurement results, thereby enhancing product quality and production efficiency. When utilizing a three-dimensional optical scanning measurement system for product measurement, the system can be positioned in an appropriate location to capture images and data of the object using a scanner or camera, as depicted in Fig. 3. The specific process unfolds as follows:

Preparation: Identify the spray developer parts to be measured and ensure the measurement system is properly prepared.

System installation: Position the three-dimensional optical scanning measurement system in the desired location, considering stability and complete coverage of the object to be measured.

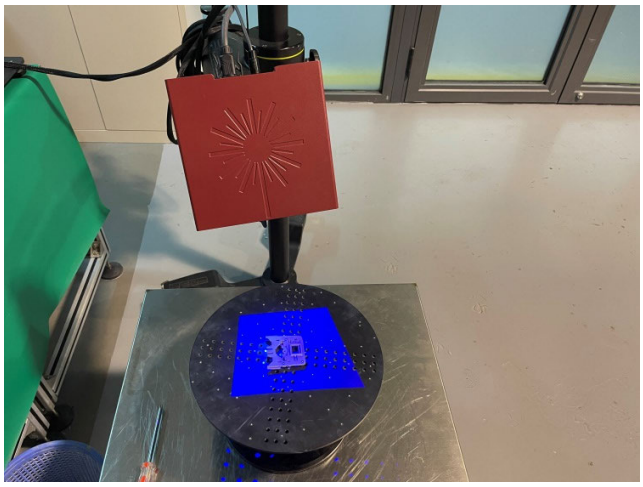


FIGURE 3. 3D optical scanning measurement system.

Parameter setup: ConFig. the scanner or camera parameters according to the measurement requirements, including resolution, scanning strategy, and other relevant settings.

Scanning measurement: Activate the three-dimensional optical scanning measurement system and initiate the scanning process for the spray developer parts. The system employs lasers or light to scan the object's surface, capturing three-dimensional coordinate data. Data processing: Import the acquired three-dimensional coordinate data into dedicated software for processing. Manipulate the point cloud data and perform shape reconstruction or other necessary operations as required.

Analysis and evaluation: Utilize the processed data to conduct comprehensive analysis and evaluation of the spray developer parts, focusing on dimensions, shapes, and other pertinent aspects. Comparisons can be made against design specifications or established standards to assess any deviations during the manufacturing process.

In this paper, Section II-A1 provides an insightful exposition on the process of spraying developer and the potential defects that may arise from it. Among the various defects, powder detachment, wear, and scratch defects are particularly noteworthy due to their significant impact on the scanning results. These two types of defects can be categorized as instances of missing developer on the surface of the part. The repercussions on the scanning data can be summarized as follows:

Data incompleteness: The scanner fails to capture the point cloud data in the areas where the developer is missing. Consequently, this leads to incomplete point cloud data, resulting in a lack of surface information and potential omission of the object's geometric shape and distinctive features.

Localized anomalies: The presence of missing areas can give rise to anomalies in the surrounding point cloud data. For instance, the point cloud data around the missing

regions may exhibit fractures, deformations, or other irregular patterns.

Impaired surface accuracy: The existence of missing regions can compromise the accuracy of the surface data. With a lack of surface information in these regions, the point cloud data surrounding them may incur substantial measurement errors, thereby diminishing the overall precision and accuracy of the point cloud data.

Visualization and post-processing challenges: The missing regions can present difficulties in visualizing and post-processing the point cloud data. In terms of visualization, the absence of data in these regions can result in visual voids and discontinuities when rendering or displaying the point cloud data. Moreover, the presence of missing regions can impede tasks such as segmentation, alignment, or surface reconstruction when performing post-processing on the point cloud data.

Therefore, in 3D optical scanning measurement systems, the absence of part developers can give rise to various challenges, including incomplete point cloud data, localized anomalies, compromised surface accuracy, and issues with visualization and post-processing. In practical applications, it is essential to employ suitable processing methods to address the missing regions. These methods may involve compensatory acquisition using other sensors, interpolation techniques to fill in the gaps based on available data, or the utilization of advanced data processing algorithms. Implementing such approaches will enhance the completeness and accuracy of the point cloud data, thereby improving the overall reliability of the scanning results.

The experiment utilized the GOM ATOS Core 200 equipment, with GOM Scan 2018 serving as the data acquisition software and GOM Inspect 2018 as the data processing software. The scanning results depicting the missing developer can be observed in Fig. 4. Additionally, Fig. 5 showcases the complete repair process of the scanned missing data using the cross-complementary hole function of GOM Inspect 2018, depicted from steps a to c. For a more detailed view, the partial repair process for the missing data scan is demonstrated in Fig. 6.

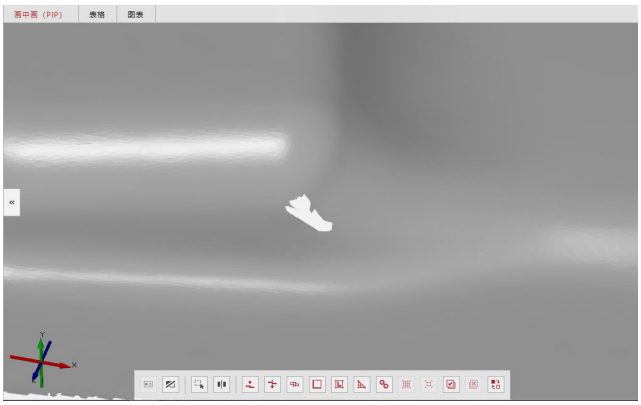
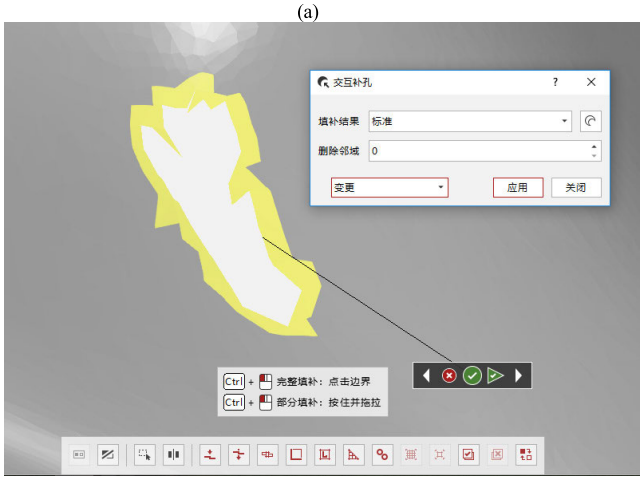
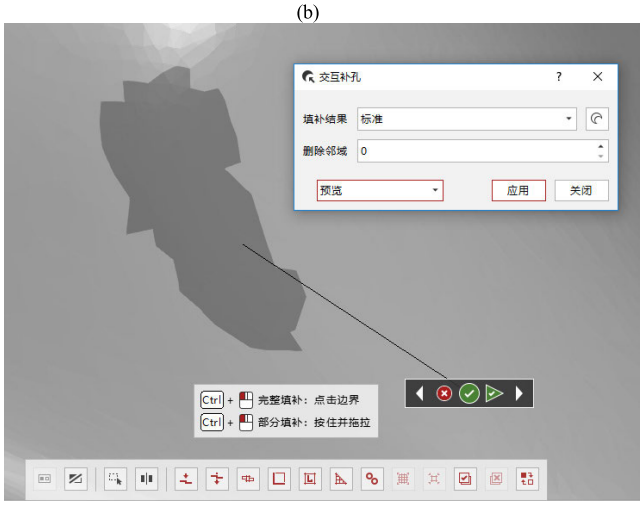


FIGURE 4. 3D scanning results of developer deficiency.

(a)



(b)



(c)

FIGURE 5. Full repair process of scanning missing data using the cross-complementary hole function of GOM Inspect 2018 software (from a to c): (a) Prototype state with missing data (b) Change status of missing data (c) Repair status of missing data.

From Fig.s 5 and 6, it is evident that the intersecting patch hole function proves effective in filling the missing regions within the 3D point cloud or surface data, thereby providing a comprehensive model. However, it is important to acknowledge that the hole-filling process may introduce certain data errors, including:

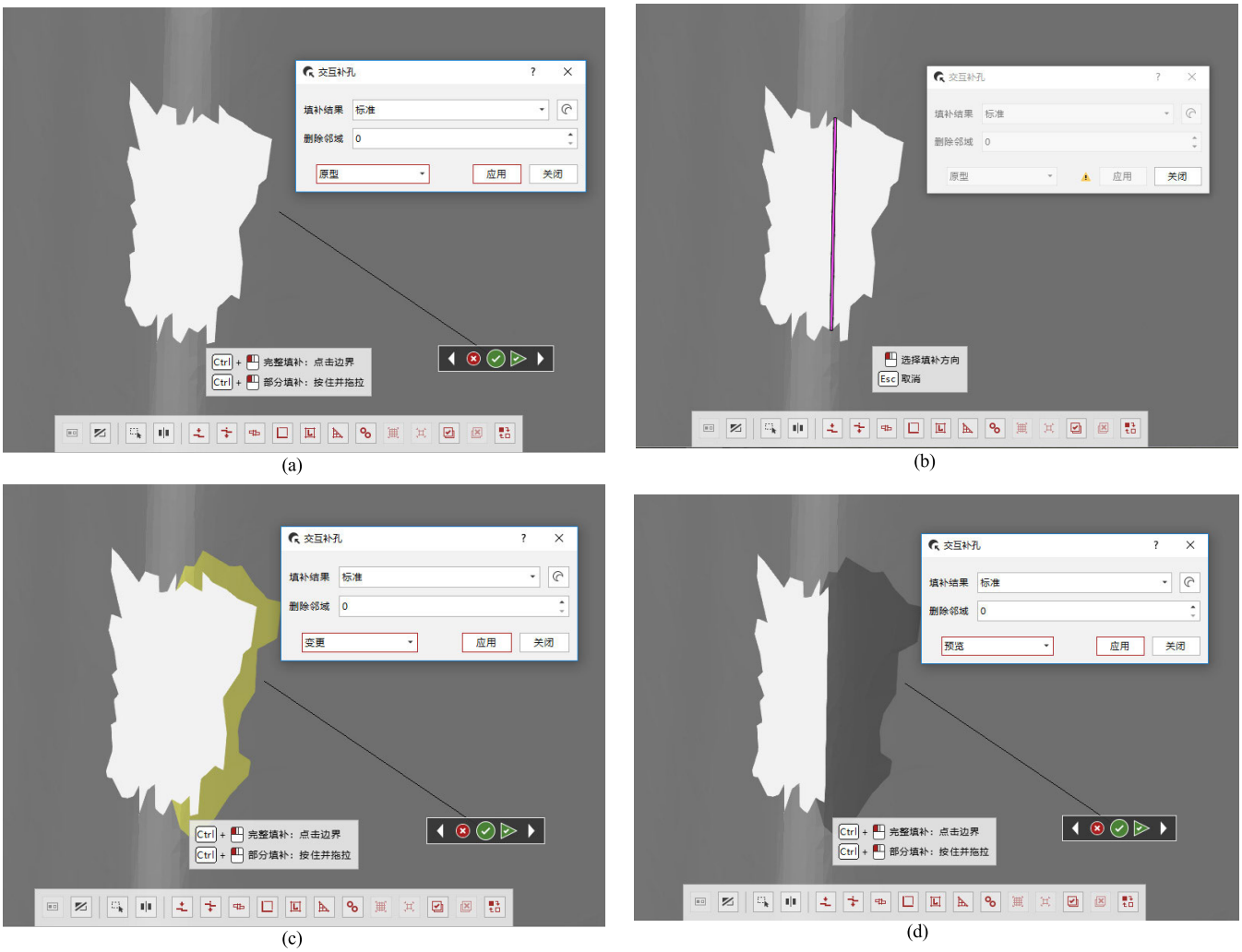


FIGURE 6. Full repair process of scanning missing data using the cross-complementary hole function of GOM Inspect 2018 software (from a to d): (a) Prototypical shape of missing data (b) Draw the data fill hole demarcation line and select the right side as the filling direction (c) Change status of missing data (d) Status of repair of missing data.

Shape distortion: During the hole-filling process, the algorithm estimates the shape of the missing region based on the surrounding data. As a result, there may be slight deviations in the shape of the patched region compared to the actual situation, leading to shape distortion.

Data smoothing: Patch hole algorithms commonly employ interpolation methods like nearest neighbor interpolation or surface fitting to fill in the missing regions. This can result in the data within the patched area being smoothed, potentially causing discrepancies in smoothness compared to the surrounding data and introducing errors due to data smoothing.

Data consistency: Since the missing area cannot be directly scanned for actual measurements, the patched data may exhibit some variation from the true situation. Consequently, the consistency between the data in the patched area and the surrounding data may decrease.

Numerical errors: The computation and interpolation processes involved in hole-patching algorithms can introduce numerical errors. These errors may stem from the algorithm's implementation details, such as calculation accuracy and choice of interpolation method, potentially impacting the data after hole patching.

Indeed, various scanning and data processing software, similar to the GOM software, offer their own hole-filling functions. The effectiveness and potential errors associated with these functions depend on the software's algorithms, parameter settings, and the characteristics of the processed data. Therefore, when utilizing the hole-filling function, it is advisable to carefully evaluate and adjust it according to the specific application requirements and data quality standards. For applications demanding high precision and accuracy, alternative data processing methods or the collection of more comprehensive raw data may be necessary to minimize the impact of errors. Currently, one approach to acquiring more comprehensive raw data involves reducing the occurrence of sprayed developer defects. However, relying solely on manual observation for defect detection may be suboptimal.

Therefore, the development of a reliable defect detection method is essential to improve the process.

B. SET ENHANCING THE ACQUISITION OF SPRAY DEVELOPER DEFECT DATA SETS THROUGH INTELLIGENT ALGORITHMS

1) OPTIMIZING THE COLLECTION OF DEFECT DATA SETS

To cater to the unique requirements of spray developer defect detection, a novel image acquisition system named EasyGit (as shown in Fig. 7) has been developed. This system incorporates a flexible camera (enclosed within the red box in Fig. 7) that offers exceptional adaptability. During the dataset collection process, a 5000-lumen 60W white LED light was positioned 2 meters above the workpiece as the light source. The image capture system, EasyGit, was placed at a distance of 20 to 50 cm from the workpiece and adjusted to capture images from various angles and directions. A total of 2000 images of spray developer defects were collected, and all images were saved in the JPEG format.

FIGURE 7. Image acquisition system EasyGit.

2) DEFECT DATA PREPROCESSING

Data augmentation is a widely employed data preprocessing method in the realm of deep learning. Its primary objective is to address the challenge of limited data by expanding the dataset. By doing so, the aim is to reduce the risk of model overfitting and enhance the overall robustness of the model. This approach helps to mitigate the overfitting problem that arises due to insufficient datasets. In the context of this paper, the experimental database is further expanded through the utilization of techniques such as Gridmask, ShearX(Y), and other augmentation methods. The database expansion process is visually illustrated in Fig. 8.

a: INTRODUCTION TO THE GRIDMASK METHOD

GridMask is a technique employed to create a mask that matches the resolution of the original image. This mask is then multiplied with the original image, resulting in a GridMask-enhanced image [40]. In Fig. 9, the gray area

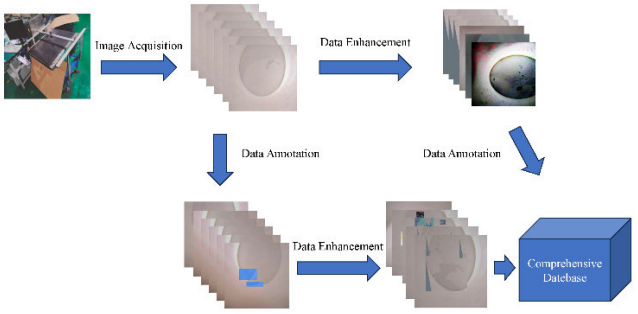


FIGURE 8. Database expansion process.

represents a value of 1, while the black area represents a value of 0. Multiplying the mask with the original image effectively achieves information dropping in a specific area.

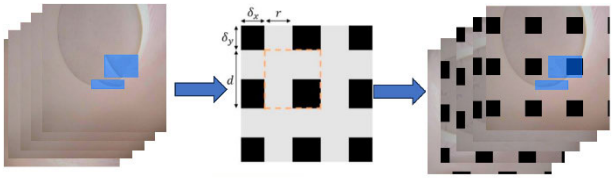


FIGURE 9. Gridmask enhancement of image process [40].

A GridMask is defined by four parameters: x, y, r, and d, which collectively determine a specific set of masks. During practical application, the mask is also subjected to rotation. The x and y values introduce a degree of randomness, while r represents the proportion of the original image that is retained. The parameter d determines the size of the dropped square. The definition of parameter r can be converted using the k value.

$$k = \frac{Sum(M)}{Hw} \tag{1}$$

$$k = 1 - (1 - r)^2 \tag{2}$$

$$\delta_x(\delta_y) = random(0, d - 1) \tag{3}$$

To begin, let us define the parameter k, which represents the retention ratio of image information. This retention ratio is calculated based on the height (H) and width (W) of the original image, as well as the number of pixels (M) that are retained. The retention ratio k is determined by these variables, and it should be noted that this parameter is not directly linked to the four aforementioned parameters. However, these four parameters indirectly define the value of r, which influences the retention ratio k.

b: INTRODUCTION TO THE SHEARX(Y) METHOD

ShearX(Y) is a fundamental process involving affine transformation along the x/y axis while keeping the other axis constant. This transformation is visually depicted in Fig. 10.

In ShearX, the input parameter θ represents the angle used for the affine transformation. It should be noted that θ is confined within the range of ($-\pi/2$, $\pi/2$). For a given coordinate point (x, y) relative to the center point (x_c, y_c),

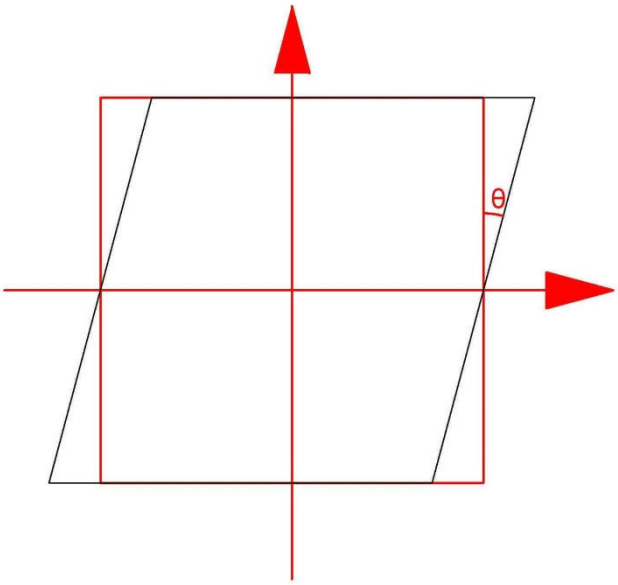


FIGURE 10. The process of ShearX(Y) affine transformation.

the final position of the point after the ShearX transformation can be determined as follows:

$$x' - x_c = (x - x_c) - tan(q)(y - y_c) \quad (4)$$

$$y' - y_c = y - y_c \quad (5)$$

In the ShearY transformation, the input parameter θ also lies within the range of ($-\pi/2$, $\pi/2$). However, in this case, it affects the transformation along the y-axis. Consequently, for a given coordinate point (x, y), the final position of the point after the ShearY transformation can be determined as follows:

$$x' - x_c = x - x_c \quad (6)$$

$$y' - y_c = (y - y_c) - tan(q)(x - x_c) \quad (7)$$

The outcomes of applying the Shear transformation to the database are visually presented in Fig. 11.

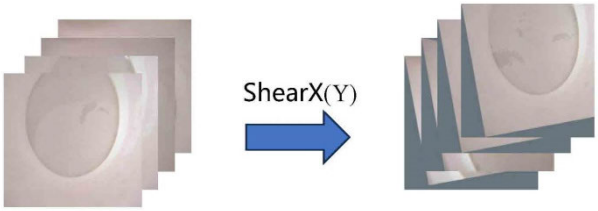


FIGURE 11. Shear transformation results for the database.

C. DEVELOPING AN INTELLIGENT ALGORITHM FOR SPRAY DEVELOPER DEFECT RECOGNITION

1) YOLO V7 ALGORITHM

The YOLO (You Only Look Once) v7 algorithm is a remarkable single-stage approach for target detection. Fig. 12 showcases the network structure of YOLO v7 [39], providing a visual representation of its architecture. To enhance the detection of small targets, a fusion of the YOLO v7 model preprocessing method with YOLO v5, utilizing mosaic data augmentation, is employed. The architecture introduces a novel extension known as Extended ELAN (E-ELAN) [39], which builds upon the ELAN concept. E-ELAN incorporates extension, shuffle, and merge bases to continuously enhance the network's learning capabilities while preserving the integrity of the original gradient paths. The computational block architecture employs group convolution to expand the channels and bases of computational blocks, allowing different groups of computational blocks to learn a broader range of features. Additionally, there are optimization modules and methods known as trainable "bag-of-freebies" [39], which include the following:

The re-parameterized convolutional architecture was designed using Identity-free RepConv. This decision was made because the presence of identity in RepConv may interfere with the residual structure in ResNet and the cross-layer connectivity in DenseNet. By eliminating the identity, we can preserve gradient diversity for various feature maps, leading to enhanced performance [41].

Auxiliary Detection Head: Deep supervision is a widely-employed technique for training deep neural networks. Its core idea is to integrate supplementary auxiliary heads into the intermediate layers of the network, along with shallow network weights guided by auxiliary losses. In YOLO v7, we introduce the Auxiliary Detection Heads, which leverage soft labels generated during the optimization process for both the Lead Head and Auxiliary Head learning. Consequently, these soft labels are anticipated to better capture the distribution and correlation between the source data and the target, thereby enhancing the accuracy of the obtained results.

Exponential Moving Average (EMA) is a technique utilized during the inference stage to incorporate the batch normalization mean and variance into the biases and weights of convolutional layers. This integration guarantees that the batch normalization mean and variance, which are averaged over batches, are effectively merged into the biases and weights of convolutional layers during the inference phase [42].

In YOLOR, the fusion of implicit knowledge with convolutional feature maps is achieved through a multiplication-based approach. This method simplifies the computation values into a vector during the inference stage, which can then be combined with the biases and weights of either the preceding or succeeding convolutional layers [43].

2) IMPROVED INFERENCE MECHANISM OF THE YOLO V7 ALGORITHM

Based on the low density of the inspected parts (specifically, sheet metal parts) in the experiment and the requirement for real-time defect statistics, the YOLO v7 model is selected in this paper. The slice-assisted hyper-inference [44]

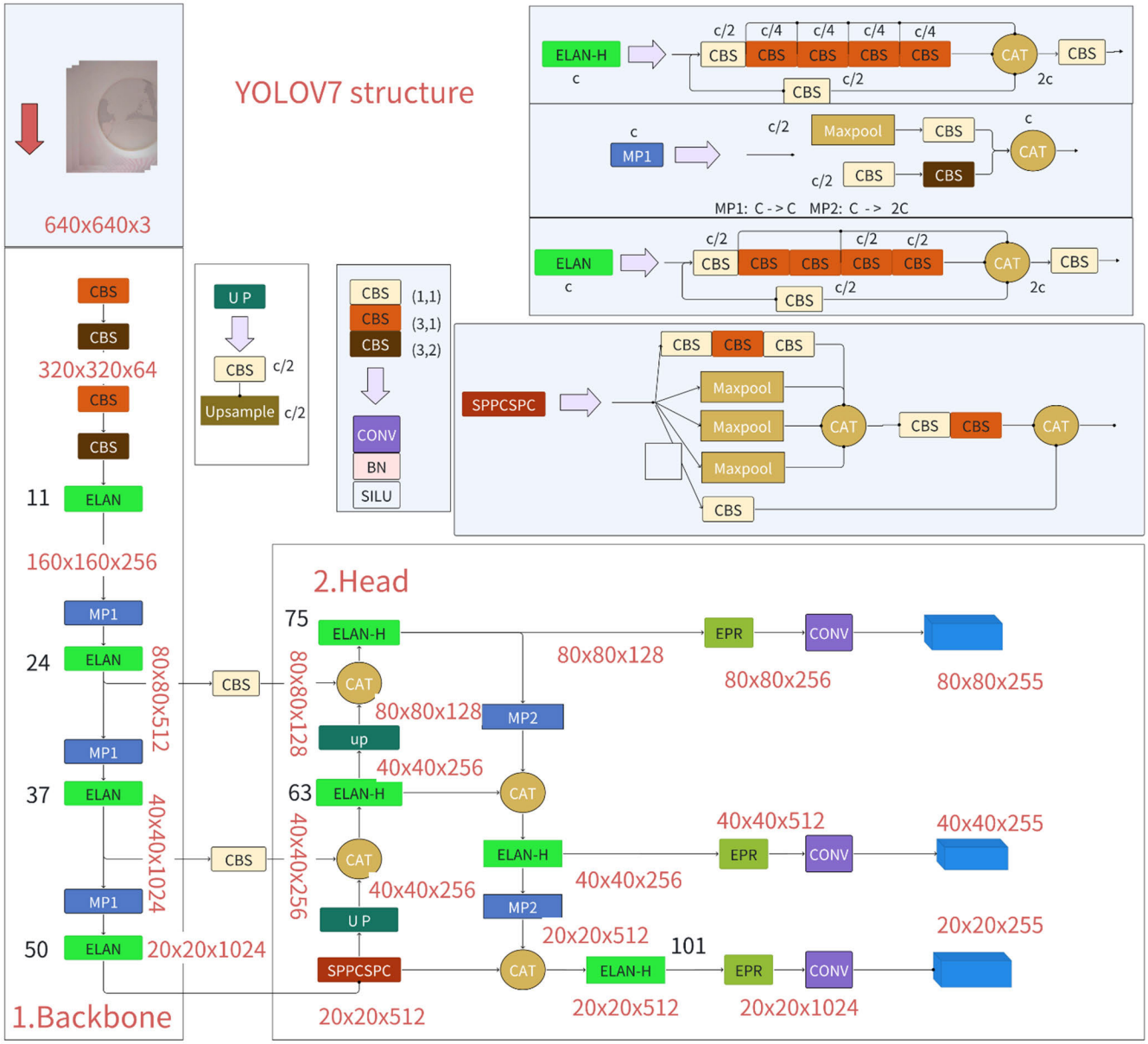


FIGURE 12. Network structure diagram of YOLO v7.

offers a comprehensive pipeline for small target detection, addressing the limitations faced by traditional detectors in detecting a small number of pixel representations and images lacking sufficient detail. To address the challenge of detecting smaller defects that are difficult to identify in real production settings without retraining the model specifically for small targets, this paper introduces a module into the YOLO v7 inference section based on the following principle:

In the inference step, a slicing approach is employed (as shown in Fig. 13). Initially, the original query image, I, is divided into L M × N overlapping patches: $P_1^I, P_2^I, \ldots P_t^I$. Each patch is then resized while maintaining the aspect ratio. Subsequently, object detection forward passes are independently applied to each overlapping patch, and for detecting larger objects, optional full inference (FI) using the original image can be conducted. Finally, the overlapping prediction results and the FI results are combined back to their original size using Non-Maximum Suppression (NMS). During NMS, boxes with intersection over merge (IoU) ratios above a predetermined matching threshold, Tm, are matched. For each match, detections with detection probabilities below Td are eliminated.

This paper employs the algorithm to achieve real-time detection of defects and their types in inspected sheet metal parts. The detected defects are used to guide the repair of

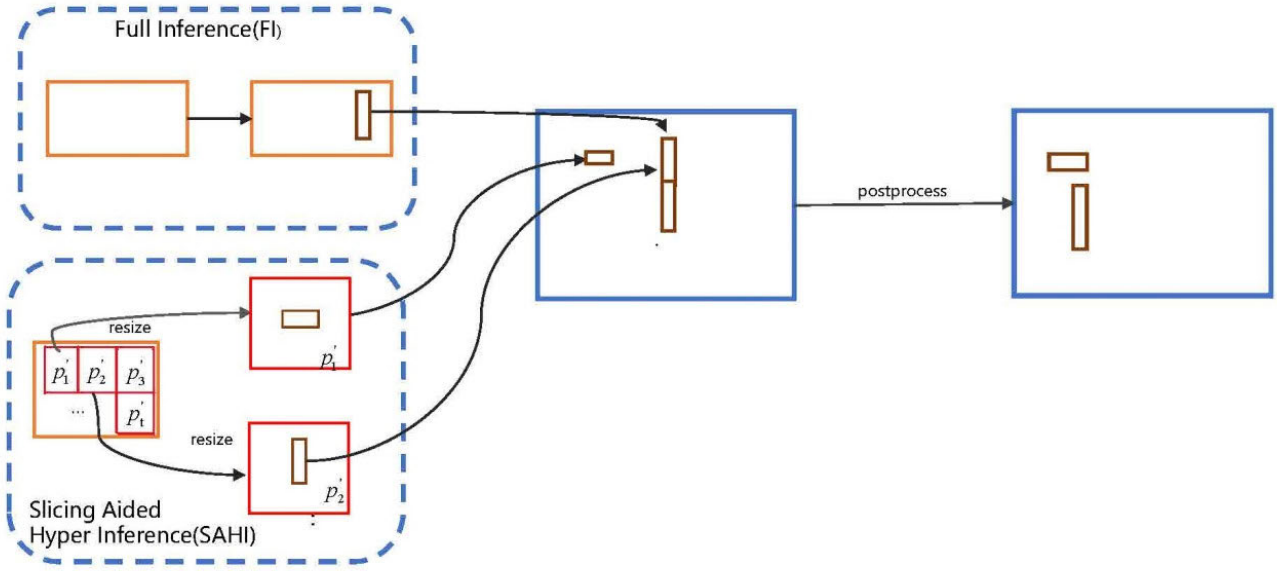


FIGURE 13. Reasoning process based on improved inference mechanism YOLO v7 algorithm.

the defects in the sprayed developer of the parts, ensuring the acquisition of complete and accurate data for subsequent 3D optical scanning. Additionally, this approach provides technical support for the automated management of the 3D optical scanning process for the inspected sheet metal parts.

D. TRAINING PARAMETERS FOR INTELLIGENT ALGORITHM

In order to demonstrate the effectiveness of the algorithm proposed in this paper, the Beijing Super Cloud Computing platform was utilized. The algorithm was developed using the Anaconda environment, with the PyTorch 1.13.1 deep learning framework running on a compute GPU, specifically the NVIDIA GeForce RTX 3090. The images were resized to a dimension of 640 pixels × 640 pixels as the input. The hyper parameters for the training of the detection algorithm are specified in Table 1.

TABLE 1. Hyper parameter settings.

Learning Rate	Batch Size	Image Size	Weight Decay	Momentum	Epochs	optimizer	Learning Rate Decay Type
0.01	24	640×640	0.0005	0.973	300	SGD	Cos

The choice of the SGD optimizer was based on previous experiments, which showed that it achieved higher accuracy compared to the ADAM optimizer. The ADAM optimizer has the advantage of effectively utilizing adaptive learning rates and momentum adjustments, making it more efficient for large-scale datasets. However, its adaptive mechanisms may lead to performance degradation on smaller datasets. In contrast, SGD demonstrates better generalization and overfitting resistance [45]. Since our dataset is relatively small, the SGD optimizer is more suitable.

The dataset was generated in COCO format [46] and labeled using the Labeling tool. Data augmentation techniques were applied to expand the dataset, as mentioned earlier. After manual selection and data augmentation, the original set of 2000 images was reduced to a dataset of 1500 images. The dataset was randomly divided into a training set and a validation set in a 4:1 ratio.

Furthermore, K-means clustering [47] was initially used on the COCO dataset to obtain predefined bounding box sizes for YOLO v7. However, considering the significant differences between the COCO dataset and our spray developer defect dataset, an improved K-means clustering method was employed to cluster our spray developer defect dataset. This allowed us to obtain anchor box sizes that are suitable for our research. The results are presented in Table 2 and were used as the predefined anchor box sizes for all algorithms used in this study.

TABLE 2. Defaulted sizes of anchor boxes.

Feature Map	640×640
Sizes clustered by K-means	[[(40, 49), (61, 69), (95, 57)], [(80, 108), (144, 92), (108, 167)], [(165, 150), (295, 135), (223, 260)]]

In order to evaluate the performance of the algorithm, the evaluation metric used in this study is the mean Average Precision (mAP) at different thresholds. The Average Precision (AP) is the average of the highest precision at different recall levels (usually calculated separately for each class), as

in (8).

$$AP = \frac{1}{11} \sum_{0,0.1...1.0} P_{smooth}(i) \tag{8}$$

The total loss function of YOLO v7 consists of three components: classification loss, localization loss, and confidence loss. When the IoU threshold is set to 0.5 [48], if multiple detections occur for an object, the object with the highest confidence is considered as a positive sample, while the other object is considered as a negative sample. On the smoothed PR curve, precision values are obtained at 10 equidistant points (including 11 breakpoints) on the horizontal axis from 0 to 1, and the average value is computed as the final AP value.

The mean Average Precision (mAP) is the average of the Average Precision (AP) values for each class, representing the average precision across all classes. It is computed by taking the average of the AP values, as in (9).

$$mAP = \frac{\sum_{j=1}^{S} AP(j)}{S} \tag{9}$$

Here, S denotes the total number of classes, and the numerator corresponds to the summation of AP values across all classes. Since this study focuses solely on the detection of a specific type of powder coating defect, AP is equivalent to mAP.

III. RESULTS AND DISCUSSION

A. COMPARATIVE ANALYSIS OF DETECTION PERFORMANCE OF DIFFERENT ALGORITHMS

According to the results presented in Table 3, the YOLO v7 algorithm outperforms the other tested detection algorithms in terms of overall performance. For instance, the mean Precision-averaged Score (mAPS) of the YOLO v7 algorithm is 6.74% higher than that of YOLO v5 [49], and the remaining metrics were either superior to or comparable with other object detection algorithms. Therefore, it can be concluded that YOLO v7 is a suitable choice as the target detection algorithm for the experiment.

TABLE 3. Comparison of results from each algorithm.

Method	mAP50	mAP75	mAPS	mAP	FPS
SSD	89.33%	70.50%	66.60%	61.40%	32.4
Faster-Rcnn	96.80%	88.30%	81.90%	79.50%	5.3
Efficientent	78.10%	34.90%	57.50%	40.20%	12.1
Deformable-detr	86.30%	54.30%	57.70%	47.30%	8.5
YOLOv5	98.00%	96.80%	92.30%	91.50%	30.1
YOLOv7	99.84%	99.60%	99.04%	99.80%	28.1

B. COMPARATIVE ANALYSIS OF DETECTION RESULTS BETWEEN TRADITIONAL ALGORITHM AND YOLO V7 ALGORITHM WITH IMPROVED INFERENCE MECHANISM

In practice, the detection of certain points becomes challenging due to external environmental interference and the random sizes of defects generated during the spraying process, as depicted in Fig. 14. In the industry, small targets are typically detected by re-capturing the image and retraining the model. However, this approach is time-consuming, labor-intensive, and does not align with actual production requirements. It also increases the operational costs for enterprises and fails to meet the demands of sustainable development. To address these issues, this paper proposes an improved inference mechanism based on the YOLO v7 algorithm for target detection. The target detection results, as shown in Fig. 15, demonstrate that the inclusion of slice-assisted inference in the inference process of YOLO v7 effectively resolves the aforementioned problems. This approach significantly reduces the influence of external factors on the processing, thereby enhancing the overall detection performance.

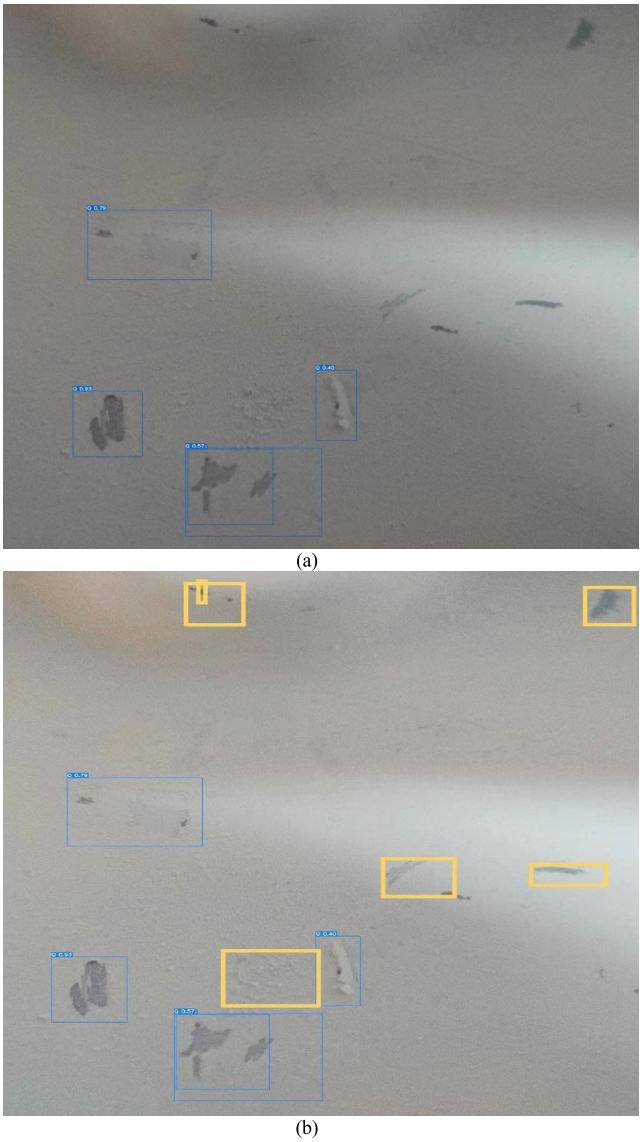


FIGURE 14. Target detection results of the traditional algorithm: (a)The actual detection results (b)Some small targets are lost as marked by yellow boxes.

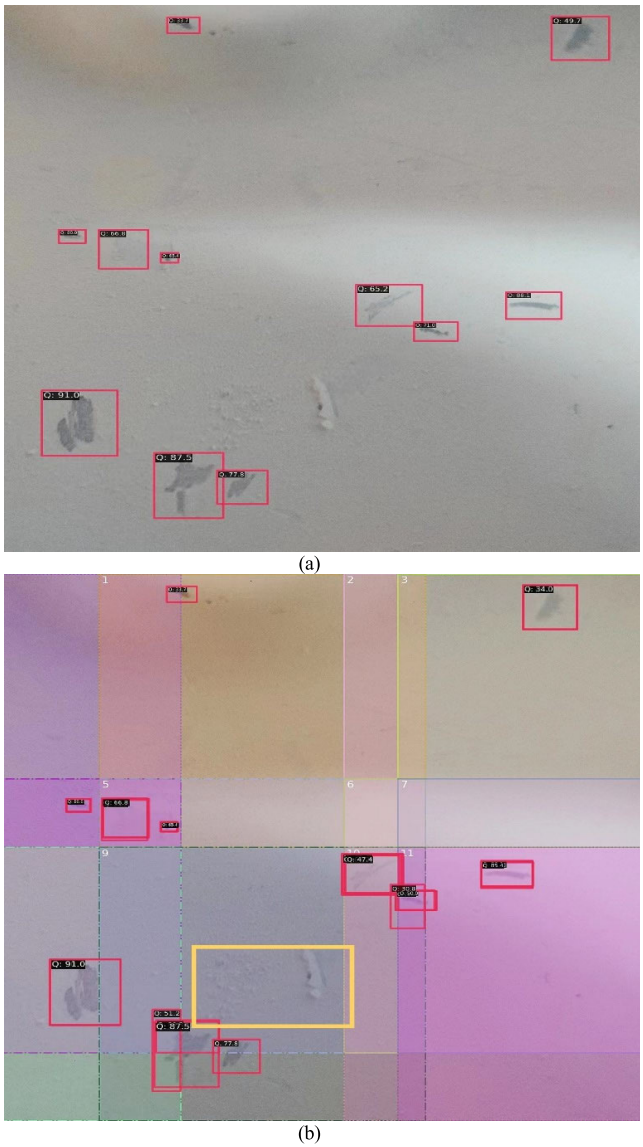


FIGURE 15. Target detection results based on the improved inference mechanism YOLO v7 algorithm: (a) The detection results (b) An example of slices put together after displaying the slices.

C. COMPARATIVE ANALYSIS OF SCANNED POINT CLOUD DATA: SHEET METAL PARTS WITH DEFECTIVE SPRAYED DEVELOPER VERSUS REPAIRED SHEET METAL PARTS

The sheet metal parts, one with defects caused by sprayed developer and the other after defect repair, were scanned using a GOM ATOS Core 200 3D optical scanning system. The 3D scanning results of the sheet metal part with spray developer defects detected by the method proposed in this paper are presented in Fig. 16. Similarly, Fig. 17 displays the 3D scanning results of the sheet metal part after the defects have been repaired. Additionally, Table 4 compares the scanned point cloud data of the sheet metal part with spray developer defects to that of the sheet metal part after defect repair. Upon analyzing Fig. 16, Fig. 17, and Table 4, it becomes evident that scanning the defective sheet metal parts with sprayed developer leads to the presence of holes, which adversely affects the accuracy of product detection. However, by applying the defect detection technique proposed in this paper, the number of holes in the sheet metal parts after defect repair has decreased by 77.2%, while the number of points has increased by 1.2% compared to the defective sheet metal parts with sprayed developer. These findings demonstrate that the method described in this paper effectively detects defects, reduces the number of holes, and increases the amount of point cloud data for the workpiece. As a result, it provides high-quality raw data for subsequent product inspection.

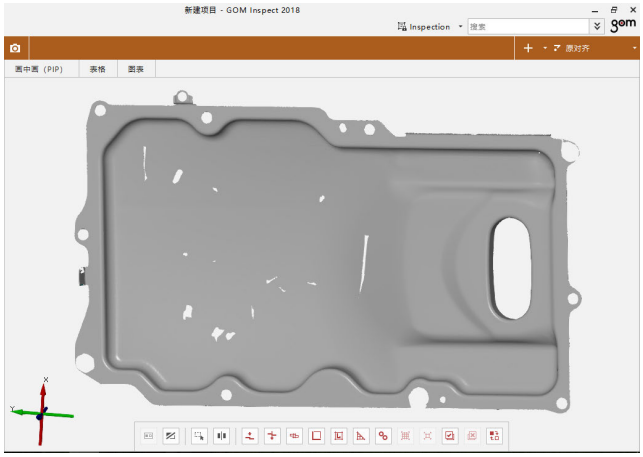


FIGURE 16. 3D scanning results of sheet metal part with defective sprayed developer.

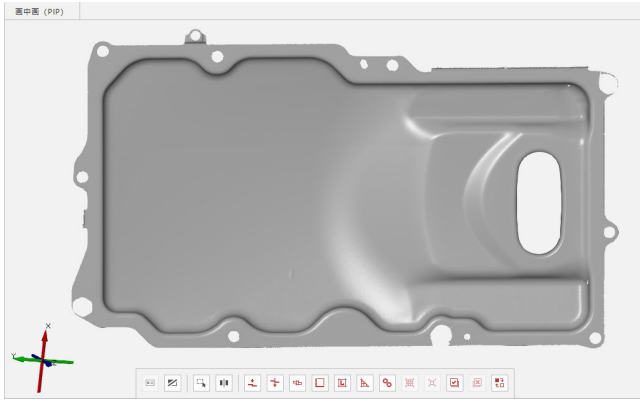


FIGURE 17. 3D scanning result of sheet metal part after defect repairing.

TABLE 4. Comparison of scanned point cloud data of sheet metal parts with defective sprayed developer and sheet metal parts with defective repairs.

Types of sheet metal parts	check numbers	hole
Defective part	1526839	145
Repair parts	1545847	33

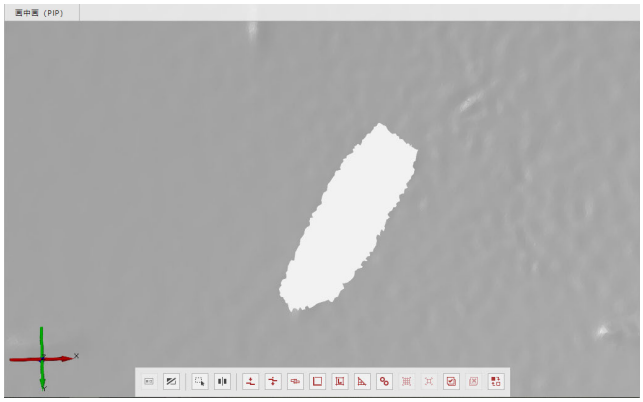


FIGURE 18. Scan data of sheet metal parts with defective sprayed developer1.

FIGURE 19. Scan data of sheet metal parts with defective sprayed developer2.

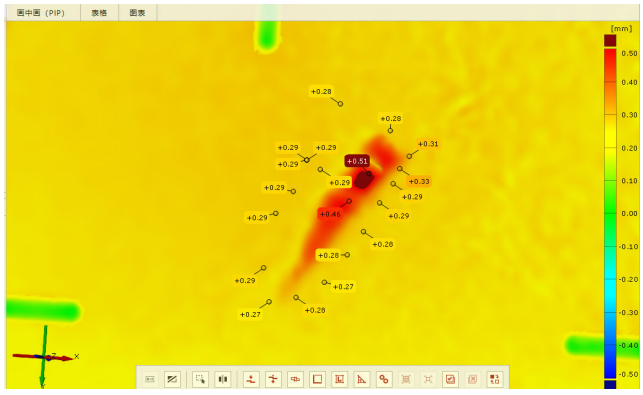


FIGURE 20. Surface inspection cloud of sprayed developer defective sheet metal part after performing. cross-complementary holes1.

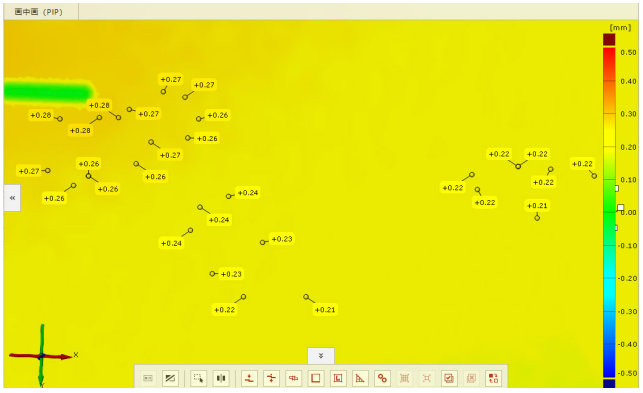


FIGURE 21. Surface inspection cloud image of sprayed developer defective sheet metal part after performing cross-complementary holes2.

FIGURE 22. Surface inspection cloud of sheet metal parts without spray developer defects1.

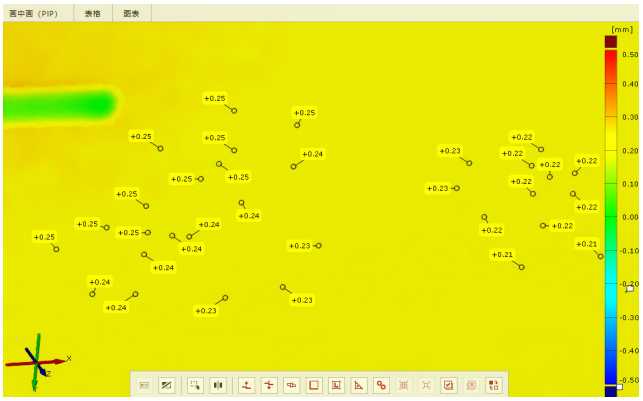


FIGURE 23. Surface inspection cloud for sheet metal parts without spray developer defects2.

D. COMPARATIVE ANALYSIS OF DIMENSIONAL INSPECTION RESULTS: SHEET METAL PARTS WITH DEFECTIVE SPRAYED DEVELOPER VERSUS REPAIRED SHEET METAL PARTS

Using the method described in section II-A2, the scan data of the sheet metal parts with spraying developer defects, as shown in Fig. 18 and Fig. 19, were intersected to patch the holes. The resulting surface detection cloud images of the sheet metal parts with spraying developer defects after hole intersection and patching can be seen in Fig. 20 and Fig. 21, while Fig. 22 and Fig. 23 display the surface detection cloud images of the sheet metal parts without spraying developer defects. Upon comparing the cloud diagrams in Fig. 20 and Fig. 22, it becomes evident that although the software can perform hole patching operations on the sprayed defects, the non-smooth transition of the point cloud data at the holes may result in a convex packet in the point cloud data after hole patching. This, in turn, leads to a significant reduction in detection accuracy, with errors of up to 70% observed in this case. However, as depicted in Fig. 21 and Fig. 23, if the transition of the point cloud data at the hole is smooth, the detection inaccuracy after hole filling can be

greatly reduced. Nevertheless, there is still a slight reduction in accuracy, with an error of approximately 8% observed in this case. By employing the spraying defect recognition technique proposed in this paper, the point cloud data of the sheet metal parts after defect repair becomes complete with smooth transitions. Consequently, the detection accuracy is significantly improved, providing more reliable results.

IV. CONCLUSION

This paper focuses on the three-dimensional scanning inspection process in the industry, specifically studying spray developer defect identification, spray developer defect target detection, and spray developer defect rejection. The aim is to provide an experimental foundation for the detection of spray developer defects.

In 3D optical scanning measurement systems, the absence of sprayed developer can result in incomplete point cloud data, local anomalies, reduced surface accuracy, and difficulties in visualization and post-processing. In practical applications, the integrity and accuracy of point cloud data can be enhanced by employing suitable processing methods for the missing regions. However, it is worth noting that existing data processing methods are prone to errors and can be time-consuming. Experimental findings have revealed that achieving complete surface coverage with developer is a direct and effective approach to improving the accuracy of point cloud data.

In this paper, we address the challenges encountered in real production scenarios by proposing an innovative inspection system. We explore different algorithms to determine the most suitable approach for detecting spray developer target uncertainty under various environmental conditions. Furthermore, we deploy the network model onto hardware devices. After careful consideration, we decide to improve the original YOLO v7 model by incorporating enhancements specifically tailored for intelligent detection of spray developer defects. This approach significantly enhances the model's target detection capabilities, improves its robustness, and ensures better adaptation to real-world scenarios. The ultimate goal is to enhance detection efficiency while minimizing manual labor costs. Through thorough evaluation, the neural network model presented in this paper achieves an impressive accuracy of up to 0.996 on the test set, and exhibits a confidence level of up to 0.96 in detecting artifacts in real-time scenarios.

In this paper, we propose the utilization of the enhanced inference mechanism of YOLO v7, which eliminates the requirement for retraining models for smaller spray developer defect targets and reduces the dependency on high-end GPUs. This advancement enables the intelligentization of spray developer defect detection, offering a viable alternative to manual detection. This approach provides an effective solution for detecting spray developer defect targets and accurate construction of 3D scanning data. It aligns with the practical demands of production and holds significant practical significance.

ACKNOWLEDGMENT

(Zhanhui Wang and Qisheng Zhao are co-first authors.)

REFERENCES

[1] J. Forest and J. Salvi, "A review of laser scanning three-dimensional digitisers," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Sep. 2002, pp. 73–78.

[2] L. Schueremans and B. V. Genechten, "The use of 3D-laser scanning in assessing the safety of masonry vaults—A case study on the church of Saint-Jacobs," *Opt. Lasers Eng.*, vol. 47, nos. 3–4, pp. 329–335, 2009.

[3] R. Mendrický and P. Keller, "Analysis of object deformations printed by extrusion of concrete mixtures using 3D scanning," *Buildings*, vol. 13, no. 1, p. 191, Jan. 2023, doi: 10.3390/buildings13010191.

[4] L. Zheng, H. Lou, X. Xu, and J. Lu, "Tire defect detection via 3D laser scanning technology," *Appl. Sci.*, vol. 13, no. 20, p. 11350, Oct. 2023, doi: 10.3390/app132011350.

[5] X. Wen, J. Wang, G. Zhang, and L. Niu, "Three-dimensional morphology and size measurement of high-temperature metal components based on machine vision technology: A review," *Sensors*, vol. 21, no. 14, p. 4680, Jul. 2021, doi: 10.3390/s21144680.

[6] A. Rekas, T. Kaczmarek, M. Wieczorowski, B. Gapinski, M. Jakubowicz, K. Grochalski, D. Kucharski, and L. Marciniak-Podsadna, "Analysis of tool geometry for the stamping process of large-size car body components using a 3D optical measurement system," *Materials*, vol. 14, no. 24, p. 7608, Dec. 2021, doi: 10.3390/ma14247608.

[7] O. Mizera, L. Cepova, J. Tkac, V. Molnar, G. Fedorko, and S. Samborski, "Study of the influence of optical measurement of slope geometry in the working chamber for AISI 316L," *Compos. Struct.*, vol. 321, Oct. 2023, Art. no. 117291, doi: 10.1016/j.compstruct.2023.117291.

[8] S. Martínez, E. Cuesta, J. Barreiro, and B. Álvarez, "Methodology for comparison of laser digitizing versus contact systems in dimensional control," *Opt. Lasers Eng.*, vol. 48, no. 12, pp. 1238–1246, Dec. 2010, doi: 10.1016/j.optlaseng.2010.06.007.

[9] J. A. Somogyi, T. Lovas, A. Szabo-Leone, and A. Feher, "Steels specimens' inspection with structured light scanner," *Periodica Polytechnica Civil Eng.*, vol. 66, no. 4, pp. 1241–1247, 2020, doi: 10.3311/PPci.20081.

[10] D. Zhang, S. N. Lingamanaik, and H. Chung, "Image-based 3D reconstruction for rail profile measurement," *Proc. Inst. Mech. Eng., F, J. Rail Rapid Transit*, vol. 237, no. 3, pp. 309–321, Mar. 2023, doi: 10.1177/09544097221110322.

[11] J. M. Navarro-Jiménez, J. V. Aguado, G. Bazin, V. Albero, and D. Borzacchiello, "Reconstruction of 3D surfaces from incomplete digitisations using statistical shape models for manufacturing processes," *J. Intell. Manuf.*, vol. 34, no. 5, pp. 2345–2358, Jun. 2023, doi: 10.1007/s10845-022-01918-z.

[12] A. Kus, "Implementation of 3D optical scanning technology for automotive applications," *Sensors*, vol. 9, no. 3, pp. 1967–1979, Mar. 2009, doi: 10.3390/s90301967.

[13] L. Hinz, S. Metzner, P. Müller, R. Schulte, H.-B. Besserer, S. Wackenrohr, C. Sauer, M. Kästner, T. Hausotte, S. Hübner, F. Nürnberger, B. Schleich, B.-A. Behrens, S. Wartzack, M. Merklein, and E. Reithmeier, "Fringe projection profilometry in production metrology: A multi-scale comparison in sheet-bulk metal forming," *Sensors*, vol. 21, no. 7, p. 2389, Mar. 2021, doi: 10.3390/s21072389.

[14] T. Várady, R. R. Martin, and J. Cox, "Reverse engineering of geometric models—An introduction," *Comput.-Aided Des.*, vol. 29, no. 4, pp. 255–268, Apr. 1997, doi: 10.1016/s0010-4485(96)00054-1.

[15] J. Zhou, Z. Ji, Y. Li, X. Liu, W. Yao, and Y. Qin, "High-precision calibration of a monocular-vision-guided handheld line-structured-light measurement system," *Sensors*, vol. 23, no. 14, p. 6469, Jul. 2023, doi: 10.3390/s23146469.

[16] Q. Zheng, B. Zou, W. Chen, X. Wang, T. Quan, and X. Ma, "Study on the 3D-printed surface defect detection based on multi-row cyclic scanning method," *Measurement*, vol. 223, Dec. 2023, Art. no. 113823, doi: 10.1016/j.measurement.2023.113823.

[17] J. Zhang, X. Wang, Y. Li, and Y. Liu, "CHBS-net: 3D point cloud segmentation network with key feature guidance for circular hole boundaries," *Machines*, vol. 11, no. 11, p. 982, Oct. 2023, doi: 10.3390/machines11110982.

[18] J. Du, H. Yan, T.-S. Chang, and J. Shi, "A tensor voting-based surface anomaly classification approach by using 3D point cloud data," *J. Manuf. Sci. Eng.*, vol. 144, no. 5, May 2022, Art. no. 0510055, doi: 10.1115/1.4052660.
[19] S. Gerbino, D. M. Del Giudice, G. Staiano, A. Lanzotti, and M. Martorelli, "On the influence of scanning factors on the laser scanner-based 3D inspection process," *Int. J. Adv. Manuf. Technol.*, vol. 84, nos. 9–12, pp. 1787–1799, Jun. 2016, doi: 10.1007/s00170-015-7830-7.
[20] N. Vukasinovic, J. Mozina, and J. Duhovnik, "Correlation between incident angle, measurement distance, object colour and the number of acquired points at CNC laser scanning," *J. Mech. Eng.*, vol. 58, no. 1, pp. 23–28, 2012.
[21] J. Forest, J. Salvi, E. Cabruja, and C. Pous, "Laser stripe peak detector for 3D scanners. A FIR filter approach," in *Proc. 17th Int. Conf. Pattern Recognit.*, Jun. 2004, pp. 646–649.
[22] M. Mahmud, D. Joannic, M. Roy, A. Isheil, and J.-F. Fontaine, "3D part inspection path planning of a laser scanner with control on the uncertainty," *Comput.-Aided Des.*, vol. 43, no. 4, pp. 345–355, Apr. 2011, doi: 10.1016/j.cad.2010.12.014.
[23] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Netw.*, vol. 61, pp. 85–117, Jan. 2015, doi: 10.1016/j.neunet.2014.09.003.
[24] Z.-Q. Zhao, P. Zheng, S.-T. Xu, and X. Wu, "Object detection with deep learning: A review," 2018, *arXiv:1807.05511*.
[25] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 580–587.
[26] H. Jiang and E. Learned-Miller, "Face detection with the faster R-CNN," in *Proc. 12th IEEE Int. Conf. Autom. Face Gesture Recognit. (FG)*, May 2017, pp. 650–657.
[27] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017, doi: 10.1109/TPAMI.2016.2577031.
[28] H. Law and J. Deng, "CornerNet: Detecting objects as paired keypoints," *Int. J. Comput. Vis.*, vol. 128, no. 3, pp. 642–656, Mar. 2020.
[29] K. Duan, S. Bai, L. Xie, H. Qi, Q. Huang, and Q. Tian, "CenterNet: Keypoint triplets for object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6568–6577.
[30] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1904–1916, Sep. 2015.
[31] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, "Deformable DETR: Deformable transformers for end-to-end object detection," in *Proc. Int. Conf. Learn. Represent.*, 2021, pp. 1–16.
[32] M. Tan and Q. V. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 6105–6114.
[33] Q. Qiu and D. Lau, "Real-time detection of cracks in tiled sidewalks using YOLO-based method applied to unmanned aerial vehicle (UAV) images," *Autom. Construct.*, vol. 147, Mar. 2023, Art. no. 104745, doi: 10.1016/j.autcon.2023.104745.
[34] D. Li, Z. Zhang, B. Wang, C. Yang, and L. Deng, "Detection method of timber defects based on target detection algorithm," *Measurement*, vol. 203, Nov. 2022, Art. no. 111937, doi: 10.1016/j.measurement.2022.111937.
[35] D. Padilla Carrasco, H. A. Rashwan, M. Á. García, and D. Puig, "T-YOLO: Tiny vehicle detection based on YOLO and multi-scale convolutional neural networks," *IEEE Access*, vol. 11, pp. 22430–22440, 2023, doi: 10.1109/ACCESS.2021.3137638.
[36] G.-Q. Wang, C.-Z. Zhang, M.-S. Chen, Y.-C. Lin, X.-H. Tan, P. Liang, Y.-X. Kang, W.-D. Zeng, and Q. Wang, "YOLO-MSAPF: Multiscale alignment fusion with parallel feature filtering model for high accuracy weld defect detection," *IEEE Trans. Instrum. Meas.*, vol. 72, pp. 1–14, 2023, doi: 10.1109/TIM.2023.3302372.
[37] J. Zheng, H. Wu, H. Zhang, Z. Wang, and W. Xu, "Insulator-defect detection algorithm based on improved YOLOv7," *Sensors*, vol. 22, no. 22, p. 8801, Nov. 2022, doi: 10.3390/s22228801.
[38] E. Dehaerne, B. Dey, and S. Halder, "A comparative study of deep-learning object detectors for semiconductor defect detection," in *Proc. 29th IEEE Int. Conf. Electron., Circuits Syst. (ICECS)*, Oct. 2022, pp. 1–2.
[39] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," 2022, *arXiv:2207.02696*.
[40] P. Chen, S. Liu, H. Zhao, and J. Jia, "Gridmask data augmentation," 2020, *arXiv:2001.04086*.
[41] X. Ding, X. Zhang, N. Ma, J. Han, G. Ding, and J. Sun, "RepVGG: Making VGG-style ConvNets great again," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2021, pp. 13728–13737.
[42] A. Tarvainen and H. Valpola, "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results," 2017, *arXiv:1703.01780*.
[43] C.-Y. Wang, I.-H. Yeh, and H.-Y. M. Liao, "You only learn one representation: Unified network for multiple tasks," *J. Inf. Sci. Eng.*, vol. 39, no. 3, pp. 691–709, May 2023, doi: 10.6688/JISE.202305_39.
[44] F. C. Akyon, S. Onur Altinuc, and A. Temizel, "Slicing aided hyper inference and fine-tuning for small object detection," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2022, pp. 966–970.
[45] N. S. Keskar and R. Socher, "Improving generalization performance by switching from Adam to SGD," 2017, *arXiv:1712.07628*.
[46] T. Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollar, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 740–755.
[47] K. Raza and S. Hong, "Fast and accurate fish detection design with improved YOLO-v3 model and transfer learning," *Int. J. Adv. Comput. Sci. Appl.*, vol. 11, no. 2, pp. 7–16, 2020, doi: 10.14569/ijacsa.2020.0110202.
[48] D. Hoiem, S. K. Divvala, and J. H. Hays, "PASCAL VOC 2008 challenge," *World Literature Today*, vol. 24, no. 1, pp. 1–4, 2009.
[49] Accessed: Aug. 15, 2022. [Online]. Available: https://github.com/ultralytics/yolov5

ZHANHUI WANG was born in Shijiazhuang, Hebei, China, in 1988. He is currently pursuing the Ph.D. degree with Yanshan University. He is currently a Lecturer with the School of Mechanical Engineering, Tianjin Sino-German University of Applied Sciences, Tianjin, China. He is a member of China Mechanical Engineering Society and China Society of Automotive Engineers. His research interests include 3D scanning inspection, reverse engineering technology, and lightweight high-strength material forming processes.

QISHENG ZHAO is currently pursuing the bachelor's degree in engineering with Tianjin Sino-German University of Applied Sciences. His research interests include deep learning and laser cladding manufacturing.

CHAOJIE FENG is currently pursuing the bachelor's degree in engineering with Tianjin Sino-German University of Applied Sciences. His research interests include 3D scanning inspection and reverse engineering.