IEEE Access
Multidisciplinary ⁞ Rapid Review ⁞ Open Access Journal

Received 28 March 2024, accepted 4 May 2024, date of publication 8 May 2024, date of current version 17 May 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3398813

RESEARCH ARTICLE

A Method for Target Detection Based on Synthetic Samples of Digital Twins

ZHE DONG, YUE YANG, ANQI WANG, AND TIANXU WU
School of Electrical and Control Engineering, North China University of Technology, Beijing 100144, China

Corresponding author: Zhe Dong (dongzhe@ncut.edu.cn)

This work was supported in part by the National Key Research and Development Program of China under Grant 2023YFC3306401.

ABSTRACT Target detection technology in the field of machine vision plays a vital role in industrial production and manufacturing. In industrial production, productivity can be improved by accurate target detection. To implement this technology, many enterprises must manually clean and label a huge dataset. Meanwhile, it is a huge challenge for enterprises to obtain the dataset because of enterprise data privacy and security constraints. This paper proposes a method for rapidly generating synthetic samples based on digital twins to address this challenge. First, the virtual environment is utilized to replicate the real detecting environment, generating a variety of sample photos. The three-dimensional coordinates of the target object are then extracted in the virtual scene. Subsequently, an annotation method is designed for synthetic samples obtained from the virtual scene, utilizing principles of three-dimensional coordinate transformation and perspective coordinate transformation. This method efficiently produces numerous labeled samples with diverse annotations. Ultimately, the model performs detection tasks in the actual world using the synthetic samples as training data. The experimental results show that the synthetic samples created by this method based on digital twins can substitute real samples and effectively identify target objects during actual detection tasks. This paper proposes a unique strategy for synthetic samples that reduces sample collection costs and privacy risks, thereby addressing the limitations of machine vision detection technology induced by sample limitations.

INDEX TERMS Digital twins, coordinates transformation, automatic annotation, synthetic samples, target detection.

I. INTRODUCTION

With the advancement of artificial intelligence technology, methods of visual examination that have been used traditionally are gradually being replaced by machine vision [1]. Within the context of the manufacturing process for industrial generation, it contributes an increasingly significant function. Intelligent manufacturing equipment, when integrated with computer vision, can utilize image recognition to determine the location of the desired object and carry out tasks such as relocation and gripping [2]. The technology of vision inspection has a wide variety of applications in the industrial sector, such as the application of robots for the sorting of materials such as wood, stone, and metal [3], as well as conducting fault identification on the surface of steel materials [4] and guided robot assembly [5]. An essential technique in industrial machine vision is Deep Learning. The real-time object detection feature of YOLO makes it possible to quickly identify and follow a variety of objects [6]. Tao et al. [7] employed the YOLO model to conduct wafer inspection in the semiconductor production process and showed that the YOLO model outperforms other models in industrial inspection. Neural network model detection frequently requires a large number of labeled real data sets, comprising data under diverse situations and annotated labels, to ensure performance.

Neural network models are trained mostly on publicly available datasets. The COCO dataset [8], supplied by Microsoft, is a comprehensive dataset used for object detection and recognition. The MVTecAD [9] is a dataset dedicated to industrial anomaly detection tasks. The

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

RSOD-Dataset [10] is a remote sensing object detection dataset that contains multiple object images. In addition, there is a data set ITODD [11] for industrial 3D target detection. The Cdiscount [12] for classifying commodity images in the Kaggle competition. The presence of the aforementioned extensive open-source datasets reduces the need for manual visual examination, enhances the precision of the visual inspection method, and simplifies the task of identifying targets.

Domain-specific detection models are required for industrial inspection, as those trained with open-source datasets are not suitable. The model's performance is also influenced by the quality of the open-source dataset. The model exhibits overfitting on these datasets, leading to inadequate generalization in real-world circumstances. Zhao et al. [13] implemented diverse data augmentation techniques to enhance the training of small sample target detection models and enlarge the sample dataset. Gautam et al. [14] proposed a migration learning strategy to tackle the issue of having insufficient datasets for training target detection models. Additionally, synthetic data can be employed to augment the sample size. Generative Adversarial Networks (GANs) [15] are models that have the ability to generate images indefinitely and are extensively employed in computer vision applications. Gao et al. [16] introduced an integrated GANs for image identification. They employed an unconditional GANs to enhance the diversity of generated images in conjunction with a conditional GANs. Recently there have been advancements in diffusion models in the domain of sample-generated data, leading to intense discussions in the field of computer vision. These models have demonstrated remarkable outcomes, particularly in scenarios with limited samples. Diffusion Models [17] offer greater visual diversity compared to GANs, and their training process is more stable. In their study, Pang and Cheng [18] adopted a target identification method that incorporated a diffusion model in order to accomplish precise detection of small targets. Taking advantage of GANs or Diffusion Models to create synthetic datasets somewhat mitigates the issue of limited sample sets. However, these models are expensive to train, necessitate substantial computational resources for executing intricate tasks in industrial settings, and yield synthetic data with subpar performance.

Object detection technology is widely used in scientific and technological life, which have great significance for sustainable development to reduce the consumption of resources in the field of machine vision. However, it is a great challenge to obtain a valid dataset for training in object detection technology. The use of publicly accessible datasets or the generation of synthetic datasets through the construction of diffusion models in previous research can alleviate this problem to some extent. Taking into account the precision of industrial inspection models, it is typically necessary for sample datasets in the industrial domain to encompass photographs captured in diverse locations, varying lighting conditions, and from many perspectives. The performance of the detection model during the collection of real sample datasets will be affected by many camera characteristics. The limitations of samples gathered in the actual world can result in a reduction in the precision of model identification. This paper presents a methodology that utilizes digital twin technology to generate samples for target detection. The method utilizes the digital twin to recreate an authentic scenario, leveraging the simulated scene to rapidly and effectively synthesize sample photos. This process results in the creation of a synthetic sample dataset, complete with labels, which can be used for target detection in the actual scene. This method of generating samples can be applied to the testing fields of industrial robotic arm grasping, object sorting, product inspection, and autonomous driving. The suggested method in this work incorporates digital twins into the target detection model, enhancing the effectiveness of the detection process and reducing the implementation cycle of the target detection model. The following are the research's primary contributions: 1) This paper present a method that enables the quick creation of labeled sample datasets for target item detection. The method efficiently retrieves the positional data of the target object from the sample image and produces a useful sample set. This set includes the synthesized sample images and a related TXT file that contains labeling information for target detection. 2) The paper introduces a technique for simulating authentic target detection settings using digital twins. The technique leverages the attributes of digital twins to efficiently produce a diverse range of samples with varying angles, distances, and circumstances, hence enhancing the effectiveness of the detection model. 3) This paper conducted research studies to assess the accuracy of target detection models trained using synthetic and actual sample datasets in a real target detection environment. This comparative experiment confirms that the synthetic samples utilized in this investigation can successfully substitute actual samples for the purpose of testing.

II. RELATED WORK

A. DIGITAL TWINS

The digital twin, initially introduced in a draft outlined by the United States in a technological roadmap for the space program [19], refers to a virtual and digitized entity that contrasts with the physical reality. The system can employ several techniques to acquire real-time data for the analysis of the physical model, monitoring, and operation and maintenance of the system [20]. The digital twin simulation scene is not limited by human senses, the virtual three dimensional(3D) model in the digital twin is easier to use and more intuitive than industrial machinery in actual industrial settings [21]. Digital twin technology is extensively employed in various domains such as the Internet of Things [22], Autonomous Driving [23], Manufacturing [24], Healthcare, and other sectors [25]. With the ongoing expansion of digital twin technology in several sectors, organizations are effectively lowering production expenses and enhancing equipment utilization during the manufacturing process.

B. SYNTHETIC SAMPLES

Bymukashev et al. [26] utilized SolidWorksCAD software to generate three dimensional models of the desired objects. The method employs this approach of constructing a bounding box around 3D objects in order to produce synthetic samples. These synthetically generated samples effectively enhance the accuracy of object detection. Damian et al. [27] employed Unreal-Engine-5 to transform real-world scene items into 3D virtual objects. They subsequently utilized the Unity Perception package [28] to produce synthetic data for the purpose of training the model. The findings demonstrate that models trained on synthetic data can be successfully extrapolated to practical real-world scenarios. Unity is a digital twin simulation program that provides a Collision volumes to GameObjects, where the Mesh Collider creates a tetrahedral mesh for any 3D model [29]. Károly et al. [30] created segmentation datasets for robotic arm vision in an automated manner. They achieved this by attaching a camera to the arm of an industrial robot and generating an object mask for each image using the camera's projection model.

III. METHOD

A. AN ARCHITECTURE FOR DETECTING SYSTEMS BASED ON DIGITAL TWIN TECHNOLOGY

This research presents a rapid and effective approach for creating accurately annotated synthetic datasets based on digital twins. The precise sequence of steps for implementing this approach is illustrated in Fig.1. The initial step involves generating a virtual testing environment of the actual testing environment. Take advantage of 3D vision sensor scanner to capture a comprehensive representation of the actual environment and obtain the point cloud model in RCP format. Next import the point cloud model into 3Dmax and subsequently crop it to generate OBJ model of the examined object and the genuine inspection environment. The integration of object models from various objects inside a physical environment to create a virtual environment for detecting purposes. The second step involves generating random sample photos with varying lighting conditions and perspectives. The digital twin's attributes are utilized to manipulate the virtual inspection environment. This involves adjusting lighting conditions, as well as the shooting angle and distance of the sampling camera. The objective is to generate a diverse collection of samples of the target object under various conditions. The third component is the automated labeling of desired objects within the generated sample photos. Acquire the coordinate data of the identified items in the simulated environment, then promptly generate the labeled synthetic dataset by manipulating and analyzing the coordinate data. Ultimately, the processed synthetic sample set is employed to train the target detection model, and the resulting model is subsequently utilized in an actual detection environment.

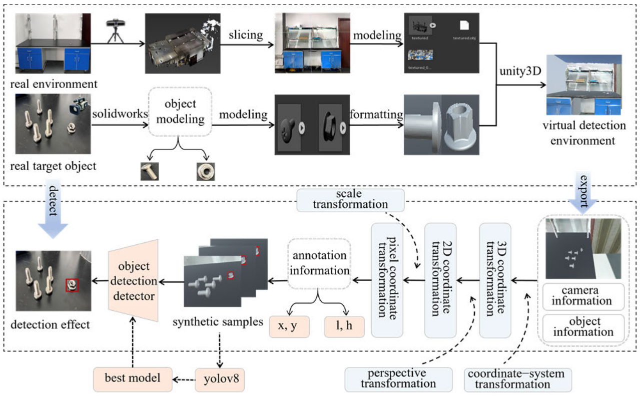


FIGURE 1. Framework of a detection system based on digital twins.

Integrating digital twins with vision inspection technology effectively addresses the problem of conducting industrial vision inspection when acquiring sample datasets is difficult or costly due to security and privacy concerns. Simultaneously, this approach can effectively circumvent the issue of imprecise labeling data for the target object resulting from dependence on manual subjective judgment for sample labeling in the detection technology. This accelerates the implementation process of inspection technology and encourages the transition of inspection instruments from theoretical simulation to quick deployment in the field of industrial automation.

B. PROCESSING METHOD FOR VIRTUAL SCENE MODELING

1) CONSTRUCTION OF VIRTUAL SCENE SIMULATIONS

The virtual testing experimental environment is constructed using the Unity engine, which includes two parts: background environment modeling and object modeling. The purpose of this environment is to detect screws and nuts on industrial parts placed on a real experimental table.

The stages of modeling the backdrop environment include: utilizing high-precision 3D scanning tools to scan environments in actual inspection environments. The scanning tool captures data from actual inspection environments and generates point cloud models in RCP format, which are easily accessible and include realistic colorful materials. At the moment, the point cloud model is often disorganized. So we can adjust the point cloud density to display the most accurate representation of the actual detection environment. The point cloud model in RCP format is imported into 3dMax to create the digital twin model of the experimental platform. The point cloud model in RCP format is imported into 3dMax to generate the digital twin model of the actual experimental platform. The polygon clipping is refined to only display the necessary area for the actual detection experimental setup. In order to create a geometric image, the cut area segmentation point cloud is used. The colorful layer has been chosen to build a simulated inspection environment. In 3dMax, a virtual inspection environment is constructed. Then the detection

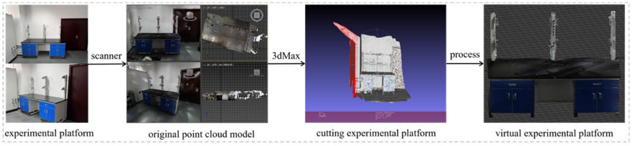
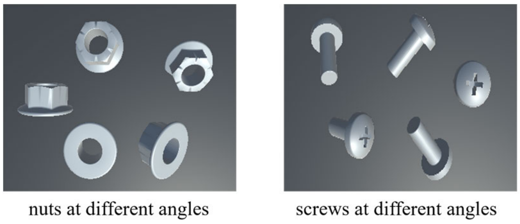

scanner
3dMax
process
experimental platform
original point cloud model
cutting experimental platform
virtual experimental platform

FIGURE 2. Processing of point cloud models.


nuts at different angles

screws at different angles

FIGURE 3. Three-dimensional model of workpieces.

environment is exported as an FBX or OBJ format of the digital twin model of the experimental bench. Subsequently, this model is imported into Unity to combine with the real inspection laboratory. The point cloud model processing flow is shown in Fig.2.

The following are the modeling steps for the object that was detected: SolidWorks is used to construct the digital twin model of the screw and nut of the observed object, creating three-dimensional textured models in the OBJ format for various workpieces. Import the model into Unity to establish hierarchical connections between objects in the inspection environment model. Assign material attributes to each individual model. Construct a mesh and construct the smallest bounding box for the 3D representation of the target object in the inspection scene. Fig.3 shows a 3D model map of various workpieces.

Once the 3D model of the detected object and the detection backdrop have been created, the detected object is positioned in the virtual detection environment by the real detection environment. To enhance the fidelity of the detected object's real context, one can incorporate additional cameras for capturing multiple angles, optimize the ambient lighting conditions, introduce realistic shadows, and implement other visual enhancements within the virtual detection environment.

2) PROCESSING OF VIRTUAL SCENE ENVIRONMENTS

Industrial target detection usually necessitates the collection of numerous samples under various circumstances. The model is trained using the sample set under varying lighting conditions and camera angles exhibits distinct performance. Typically, the greater the number of collected samples, the greater the accuracy of the trained detection model and the more precise the model's performance. The processing of virtual scenes in Unity can efficiently provide a varied sample set. This processing mostly involves modifying the lighting conditions, camera shooting angle, camera shooting position, and the position of the target object.

Algorithm 1 Obtain Diverse Synthetic Samples

Input: Initial position and lighting information
Output: Location and lighting information after changes

```
Update()
Timer + = time.delatime
If timer >= interval
RandomLightIntensity()
RandomcameraPosition ()
RandomtargetPosition ()
Timer = 0.0f
RandomLightIntensity()
a = LightIntensityChange
  newLightIntensity=initialLightIntensity+Random.Range(-a, a)
  RandomcameraPosition ()
  b = cameraRotationChange
  c = cameraPositionChange
  newcameraAngle=initialcameraRotation+Random.Range(-b, b)
  m = Random.Range(-b, b)
  Vector3random=newVector3(Random.Range(m,m,m))
  newcameraPosition=initialcameraPosition +Vector3 random
  RandomtargetPosition ()
  d = targetPositionChange
  n = Random.Range(-d, d)
  Vector3random=newVector3(Random.Range(n,n,n))
  newtargetPosition=initialtargetPosition+Vector3random
  end for
```

To render light in Unity, you can assign an initial value to the scene (initialLightIntensity). Configure a variety of light alterations (LightIntensityChange) and designate a consistent time interval for each period(interval),allowing for random changes between light and darkness within the specified range.

Similar to this, the sampling camera's parameter settings in unity are modifiable. Customizing the initial shooting angle (initialcameraRotation) and position (initialcameraPosition). The camera's change in angle (cameraRotationChange) and change in position (cameraPositionChange) can be adjusted in three dimensions, specifically around the camera's x, y, and z axis. Additionally, setting a fixed timestamp interval to control the random changes in the sampling camera's shooting angle and position based on your specific requirements.

Furthermore, it is feasible to enhance the sample set by randomly altering the location of the target object. This can be achieved by given an initial position (initialtargetPosition) and a customisable range of change (targetPositionChange), allowing for random adjustments within the specified range as needed.

As demonstrated in Fig.4, various kinds of samples can be acquired by altering the lighting conditions of the scene or adjusting the shooting angle, position, and target object position of the sampling camera. By randomly combining changes in scene lighting, shooting angle, shooting position, and target object position through algorithm 1, it is feasible to acquire multiple combinations of sample photographs, which can be used to enlarge the overall sample set.

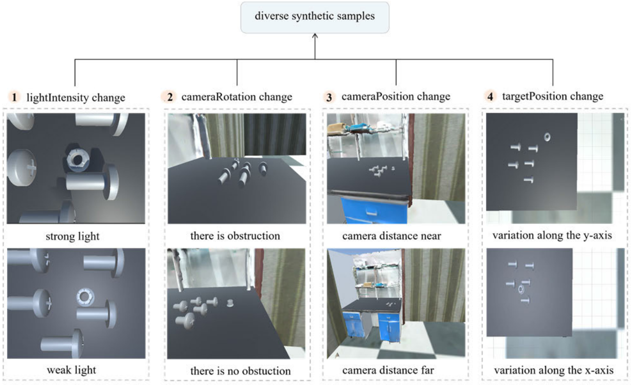


FIGURE 4. Samples under various conditions.

TABLE 1. Export data directory.

Category	Information
P_W	target object coordinates
α	target object yam angle
β	target object pitch angle
γ	target object roll angle
l	length
w	wide
h	high
P_c	camera coordinates
α_c	camera yam angle
β_c	camera pitch angle
γ_c	camera roll angle
FOV	visual cone angle
Aspect	image aspect ratio
Near	near plane
Far	far plane

3) EXPORTING VIRTUAL SCENE DATA

Numerous samples with varying conditions can be generated in a short amount of time by processing the virtual inspection environment that is modeled in Unity. A sample image export and a virtual information export are the two components that make up the acquisition of the detection sample set.

Developing programs in C# that may be attached to sample cameras. Customizing the sample image export storage path in the script, and the storage format is PNG. Every time there is a change in the environmental conditions, the virtual detection scene in Unity is able to generate a new sample image.

After exporting the sample image, as shown in Table.1. The parameter information of the target object and the sampling camera in the virtual scene can be read, according to the Inspector component in Unity. The custom information exports the storage path in the C# script and produces an identical timestamp text file for every example image. The information of the current target detection object in the virtual scene is derived. This information includes the coordinates of the center point of the target object, the angle of the target object in the x, y, and z directions (α, β, γ), as well as the length, breadth, and height of the target object (l, w, h). The information of the current sampling camera contains the coordinates of the center point P_c (x_c, y_c, z_c), and the angle in the x, y, and z directions $(\alpha_c, \beta_c, \gamma_c)$. It also involves the field of view of the camera(FOV), the aspect ratio of the camera's cone of vision(Aspect), the value of the camera's Near plane(Near), and the value of the camera's Far plane(Far). The FOV is the angle at which the camera's cone of vision gets th P_W (x_W, y_W, z_W) e picture of the object.

C. FAST LABELING METHOD FOR SYNTHETIC SAMPLES

1) ANNOTATION METHOD IMPLEMENTATION FRAMEWORK

The image processing platform processes and transforms coordinates based on the text file exported information after receiving the sample images in PNG format generated by the unity engine and the text file that corresponds to them. Following this, the target objects are labeled in the sample images for annotation. When using different reference coordinate systems, 3D object coordinate values are different.

Therefore, the processing of the coordinate of the target object is divided into two modules. First, the transformation of the target object coordinates between the three-dimensional coordinate system is done. Next, transforming the coordinates of the target item from three-dimensional space to a two-dimensional plane. The specific conversion process is shown in Fig.5. Every three-dimensional model of target detection in the digital twin has a minimal outer enclosing box that can completely encapsulate the object. The coordinate system for the target object can be established using its length, width, and height as the x, y, and z axes. The center point of the target object can serve as the coordinate origin. The eight corner of the enclosing box of the target object can be determined using the target object coordinate system. Transforming the target object coordinate system's eight corner point coordinates to the world coordinate system. Subsequently, converting the target object's world coordinate system center point and eight corner points to the camera coordinate system coordinates. As a result, the conversion of the target's coordinates from the target's own coordinate system to the camera's coordinate system has been successfully completed. The target object's eight corner coordinates and center point coordinates were then mapped to the pixel coordinates on the two-dimensional imaging plane using the idea of perspective transformation. Proportionately changing pixel coordinates to image viewport coordinates. Locate the corner in the picture that has the largest x, y coordinate and the smallest x, y coordinate among the eight corner viewport coordinates, and then label the object that you are looking for. The sample collection of labeled target objects includes a variety of example images as well as information regarding the labeling of target objects that corresponds to these sample images.

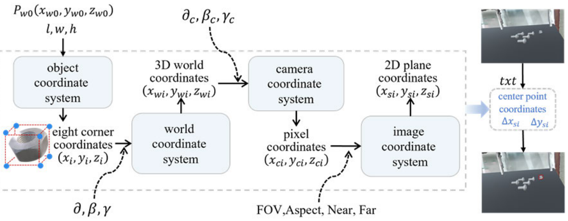

$P_{w0}(x_{w0}, y_{w0}, z_{w0})$
l, w, h
$\partial_c, \beta_c, \gamma_c$
object coordinate system
3D world coordinates (x_{wi}, y_{wi}, z_{wi})
camera coordinate system
2D plane coordinates (x_{si}, y_{si}, z_{si})
txt
center point coordinates Δx_{si} Δy_{si}
eight corner coordinates (x_i, y_i, z_i)
world coordinate system
pixel coordinates (x_{ci}, y_{ci}, z_{ci})
image coordinate system
∂, β, γ
FOV,Aspect, Near, Far

FIGURE 5. Flowchart of coordinate conversion.

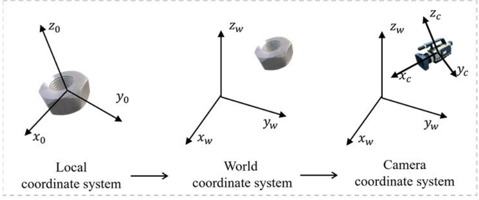


FIGURE 6. Schematic of coordinate conversion.

2) CONVERSION OF THREE-DIMENSIONAL COORDINATE SYSTEMS

As can be seen in Fig.6, the coordinates of the center point of the target object, denoted as P_W (x_w, y_w, z_w), are considered to be the coordinate origin of the target object coordinate system. The x, y, and z axes of the target object coordinate system are the length, width, and height of the target object. Finding out the coordinates of the eight corners of the box that surround the three-dimensional model of the target object in the virtual scene. These coordinates should be found under the coordinate system of the target item. According to equation:

$$y_i = \left(\pm\frac{l}{2}, \pm\frac{w}{2}, \pm\frac{h}{2}\right), i = 1, 2, 3, 4, 5, 6, 7, 8 \tag{1}$$

The coordinates of the eight corner points of the target object box should be converted from the coordinates used in the target object coordinate system to the coordinates used in the world coordinate system. In the first step of the process, the coordinate origin of the target coordinate system is transformed with the origin of the world coordinate system. The translation matrix is as follows:

$$T_w = (x_w, y_w, z_w)^T \tag{2}$$

Then rotate the x-axis, y-axis, and z-axis of the target object coordinate system to coincide with the world coordinate system, and the rotation matrix is:

$$R_w = \begin{bmatrix} \begin{matrix}\cos\beta\cos\gamma + \\ \sin\alpha\sin\beta\sin\gamma\end{matrix} & \begin{matrix}\sin\alpha\sin\beta\cos\gamma - \\ \sin\gamma\cos\beta\end{matrix} & \sin\beta\cos\alpha \\ \sin\gamma\cos\alpha & \cos\alpha\cos\gamma & -\sin\alpha \\ \sin\alpha\sin\gamma\cos\beta & \begin{matrix}\sin\beta\sin\gamma + \\ \sin\alpha\cos\beta\cos\gamma\end{matrix} & \cos\alpha\cos\beta \end{bmatrix} \tag{3}$$

The coordinates y_{wi} of the eight corner points in the world coordinate system are obtained after the mutual transformation of the coordinate system:

$$y_{wi} = R_w y_i + (x_w, y_w, z_w)^T, i = 1, 2, 3, 4, 5, 6, 7, 8 \tag{4}$$

To determine the camera coordinate system, the center point coordinates P_c (x_c, y_c, z_c) of the sample camera are used as the coordinate origin. The length, width, and height of the camera are used to determine the x-axis, y-axis, and z-axis directions of the camera coordinate system, respectively. The center point of the target object in the world coordinate system with coordinates P_W (x_w, y_w, z_w) and the eight corner points of its enclosing box in the world coordinate system with coordinates y_{wi} should be converted to the coordinates in the camera coordinate system, and then the resulting coordinates should be converted in accordance with the three-dimensional coordinate system conversion described above.

Initially, the origin of the camera coordinate system is converted into the origin of the boundary coordinate system, and the translation matrix is:

$$T_c = (x_c, y_c, z_c)^T \tag{5}$$

Then align the camera coordinate system's the x-axis, y-axis, and z-axis angles $(\alpha_c, \beta_c, \gamma_c)$ with the world coordinate system. The rotation matrix is:

$$R_c = \begin{bmatrix} \begin{matrix}\cos\beta_c\cos\gamma_c + \\ \sin\alpha_c\sin\beta_c\sin\gamma_c\end{matrix} & \begin{matrix}\sin\alpha_c\sin\beta_c\cos\gamma_c - \\ \sin\gamma_c\cos\beta_c\end{matrix} & \sin\beta_c\cos\alpha_c \\ \sin\gamma_c\cos\alpha_c & \cos\alpha_c\cos\gamma_c & -\sin\alpha_c \\ \sin\alpha_c\sin\gamma_c\cos\beta_c & \begin{matrix}\sin\beta_c\sin\gamma_c + \\ \sin\alpha_c\cos\beta_c\cos\gamma_c\end{matrix} & \cos\alpha_c\cos\beta_c \end{bmatrix} \tag{6}$$

The coordinates of the eight corner points y_{ci}in the camera coordinate system are obtained after the mutual transformation of the coordinate system:

$$y_{wi} - (x_c, y_c, z_c)^T = R_c y_{ci}, i = 1 \sim 8 \tag{7}$$

The coordinates of the center point of the target object y_{c0} are obtained after the mutual transformation of the coordinate system:

$$y_w - (x_c, y_c, z_c)^T = R_c y_{c0} \tag{8}$$

Fig.6 illustrates that it is possible to obtain the coordinates of the target object's center point and the coordinates of the eight corner points of the enclosing box of the target object in the camera coordinate system taking advantage of the inverse transformation between the three coordinate systems after the three three-dimensional coordinate systems are established.

3) THREE-DIMENSIONAL COORDINATES INTO TWO-DIMENSIONAL COORDINATES

Objects within the eye's field of view in the physical world exhibit a visual phenomenon known as ''near big and far small''. Generally speaking, when the eye is closer to the target item, the object seems larger but when the eye is farther

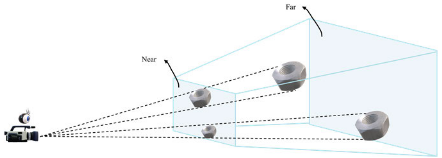


FIGURE 7. Schematic diagram of near big and far small.

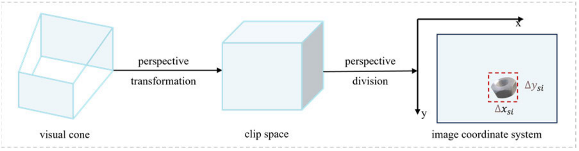


FIGURE 8. Schematic of perspective transformation.

away, the object appears smaller. Analogously, the aforementioned phenomenon arises when a camera captures an image of a certain object and subsequently acquires data from it within the unity engine. Due to the limitations of the sample camera property in Unity, the sampling range of the sample camera is actually a cropped view cone. Fig.7 illustrates the range of the sample camera, which extends from the near plane to the far plane. This range represents the actual visual range of the sampling camera in Unity. The near plane of the sampling camera can be approximated as the imaging plane of the camera.

Fig.8 demonstrates that the three-dimensional coordinates of the midpoint of the target object can be converted into two-dimensional plane coordinates. This conversion occurs when the target object, within the viewing range of the sampling camera, is compressed from the cropped view cone to the standard cube via perspective transformation.

The formula of perspective transformations is based on the perspective principle of OpenGL:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ w \end{bmatrix} = \begin{bmatrix} \frac{\cot\frac{Fov}{2}}{Aspect} & 0 & 0 & 0 \\ 0 & \cot\frac{Fov}{2} & 0 & 0 \\ 0 & 0 & \frac{Far+Near}{Far-Near} & -\frac{Far\times Near}{Far-Near} \\ 0 & 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} x\frac{\cot\frac{Fov}{2}}{Aspect} \\ y\cot\frac{Fov}{2} \\ -z\frac{Far+Near}{Far-Near} - \frac{2\times Far\times Near}{Far-Near} \\ -z \end{bmatrix} \tag{9}$$

The coordinates (x, y, z) represent the three-dimensional positions of the points on the target object in the camera coordinate system. The coordinates $(x, y, z, 1)$ represent the transformed three-dimensional positions of the points on the target object after applying the homogeneous coordinate transformation approach in the camera coordinate system. The coordinate $(\dot{x}, \dot{y}, \dot{z}, w)$ represents the position of the points on the target object in the visual range after the position of target object through perspective projection. $(\dot{x}, \dot{y}, \dot{z}, w)$ divided by the perspective viewpoint:

$$\left(\frac{\dot{x}}{w}, \frac{\dot{y}}{w}, \frac{\dot{z}}{w}, 1\right), w = -z \tag{10}$$

The homogeneous coordinates $(\dot{x}/w, \dot{y}/w, \dot{z}/w, 1)$ refer to the points of the target object in the standard cube after undergoing the perspective projection transformation. By using the aforementioned transformation, the coordinates of the eight corner points and the center point of the target object can be translated from three-dimensional coordinates to two-dimensional pixel coordinates inside the imaging coordinate system.

The scale transformations are able to make converting two-dimensional pixel coordinates to picture viewport coordinates. The image's top left corner is located at pixel coordinates $(0, 0)$, while the bottom right corner is located at pixel coordinates $(pixelWidth, pixelHeight)$. The top left corner of the image in the viewport is also at coordinates $(0, 0)$, while the bottom right corner is at coordinates $(1, 1)$. Converting two-dimensional pixel coordinates in the range of 0 to 1 to viewport coordinates. From the value$(Aspect)$ captured by the sampling camera in Unity, the screen display resolution value is known as $pixelWidth \times pixelHeight$, The scaling formula for the x and y coordinates of points in the target object:

$$x_{si} = \frac{\dot{x} \times pixelWidth}{2 \times w} + \frac{pixelWidth}{2} \tag{11}$$

$$y_{si} = \frac{\dot{y} \times pixelHeight}{2 \times w} + \frac{pixelHeight}{2} \tag{12}$$

Following the process of scaling the center point and eight corner point coordinates to the appropriate viewport range, determine the corner points' two-dimensional coordinates the $\max x_{si}$, $\min x_{si}$, $\max y_{si}$, $\min y_{si}$ and the automatic labeling frame of the target item can be obtained by using parallel lines passing through the four corner points.

The target item in the sampled picture automatically draws the labeled box in the composite sample effect image by using the unity engine to extract the information processing. Fig.9 illustrates the consequences of the unity-based sample labeling method on frame labeling for four distinct shapes of target objects: nut, multimeter, toolbox, coke, coffee and book.

IV. EXPERIMENTAL SECTION

A. DATASETS

Synthetic datasets: The dataset is a crucial factor in industrial target detection since it greatly influences the accuracy of the detection model. In the real experimental bench, there is a pile of workpiece screws and nuts, and the objective of this experiment is to differentiate and identify the nut from the rest of the pile. A synthetic sample dataset consisting of one thousand workpiece screws and nuts that were labeled was produced by the unity-based automatic annotation method, which was used to construct the training set data. Among the one thousand sample photographs that were chosen, there

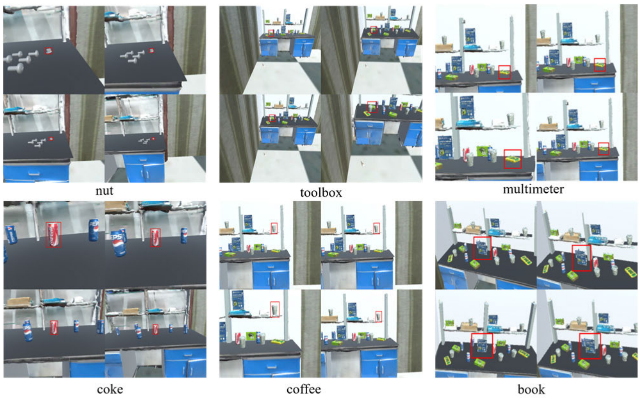


FIGURE 9. The effect demonstration diagram of the annotation.

are examples of the target item captured from a variety of perspectives and under a variety of lighting situations. Additionally, the sample images have diverse pixel resolutions. The synthetic samples that have been labeled are separated into three categories: eighty percent are classified as training datasets, 10 percent are classified as validation datasets, and 10 percent are classified as test datasets.

Real datasets: To evaluate the detection capabilities of the model trained using synthetic data in a real-world target detection scenario, we captured 1000 photos of target objects in an actual laboratory setting using a camera. Taking advantage of the LabelImg labeling method to manually annotate the target object and obtain 1000 accurately labeled real samples. Similar to the synthetic sample set, the real sample set also includes photographs of the target object captured from various perspectives and under varying lighting conditions. A dichotomous divide of the dataset is also included in this real sample, just like it was in the synthetic dataset. Eighty percent of the dataset is classified as the training dataset, ten percent is the validation dataset, and ten percent is the test dataset.

In order to investigate the consequence of using the synthetic sample set, the detection effect of the model that was trained on the actual sample set is used as an example of reference. Following the training of several target detection models using synthetic and actual sample sets, the trained models are evaluated using the same real sample test set consisting of 100 sheets. It is necessary to evaluate the performance of model detections and the efficacy of synthetic samples for real detection tasks.

B. EXPERIMENTAL SETTINGS

For the purpose of determining the impact of the synthetic sample set, the YOLOv8 model was utilized in this investigation. The YOLOv8 model is a deep learning model that integrates the most recent basic version of the YOLO family of models and introduces improvement points in order to enhance the performance of the model. A new loss function, a new detection header, and a new backbone network are all included in the YOLOv8 release. The fact that it is able to detect numerous types of objects in real-time on a variety of GPU hardware has led to this model being frequently made use of in the industrial field. The pre-training model used for all model training in this paper is yolov8 nano, which belongs to the yolov8 family of models and is known for its lightweight nature. This experiment makes use of the standard metrics for evaluating target detection models, which include precision (P), recall (R), and mean average precision (mAP). These metrics are used to describe the performance of the model:

$$P = \frac{TP}{TP + FP} \tag{13}$$

$$R = \frac{TP}{TP + FN} \tag{14}$$

$$mAP = \frac{\sum_{j=1}^{k} AP_j}{k} \tag{15}$$

Ture positive(TP) represents the count of positive samples correctly identified as positive by the model, false positive(FP) represents the count of negative samples incorrectly identified as positive, false negative(FN) represents the count of positive samples incorrectly identified as negative, and average precision(AP) represents the area under the precision-recall curve.

In order to avoid overfitting the model, the value for *PATIENCE* is configured to 50 for each training session. If the model performance does not show significant improvement after 50 rounds of iterations during the model training phase, the training is halted. The training epochs are configured to 200, the batch size is 32, and the size of the input sample images is 640 pixels ×640 pixels. Additionally, to enhance the accuracy of the model, the mosaic enhancement feature was disabled during the final 10 epochs while performing data augmentation for training. This study relies on the Pytorch inference framework for experimental training. The training and inference settings consist of Pycharm, python3.9, and CUDA11.6, and the training and inference are executed on an NVIDIA RTX 3090.

C. EXPERIMENTAL RESULTS

1) TIME COMPARISON FOR OBTAINING SAMPLE SETS

The purpose of this research is to suggest a method for the rapid acquisition of synthetic samples that can also serve as a substitute for actual sampling. The experiment included evaluating the time it took to acquire the synthetic sample set, which was generated and diversified using a sample generation method, different from the time it took to acquire the sample set through real sampling. This was done in order to determine how well the method performed. A sample generation method can produce a sample set of approximately 2000 labelled synthetic samples in a minimum of ten minutes. In the same interval of time, manual annotation can only generate about 200 labelled real sample images. Meanwhile, if manual annotation were to produce the real samples that

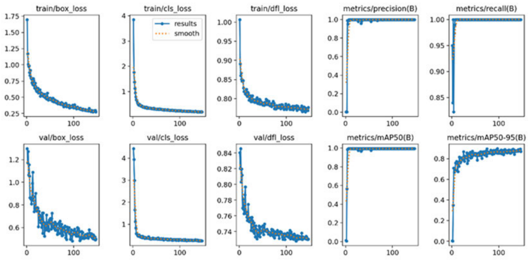


FIGURE 10. The model was trained with synthetic samples.

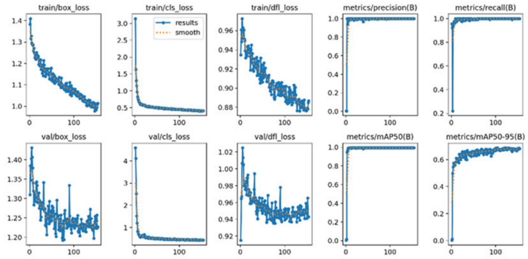


FIGURE 11. The model was trained with actual samples.

has same number of synthetic samples, it would take much longer than ten minutes. In terms of the quantity of time required to acquire a sample set, the acquisition of a genuine sample set takes significantly more time and is more challenging than the acquisition of a synthetic sample set. Furthermore, acquiring authentic samples poses challenges and incurs labor expenses.

2) PERFORMANCE EVALUATION OF DIFFERENT DETECTION MODELS

Evaluating the synthetic sample model's performance in comparison to real samples during the training process: When the synthetic samples that were created by the automatic labeling method were utilized for the purpose of pre-training the target detection model, the accuracy and recall were above 95% which is basically close to 1 after 100 rounds of model training, and the precision gradually improves with the increase of mAP 50-95 under different intersection over union(IOU) thresholds. The training process is depicted in Fig.10, and it can be seen that the loss decreases continuously as the training iteration rounds increases continuously. Furthermore, the performance parameters gradually stabilize.

The target identification model is trained on manually labeled real samples. The training process is illustrated in Fig.11. As the number of training rounds grows, the loss curve of the model exhibits fluctuations in the early stage, which then settle after 100 rounds. The target model trained by manually labeled real samples behaves normally during training.

It is possible to draw the conclusion that the performance of the model trained by the synthetic samples is essentially the same as the performance of the model trained by the real samples during the training process and that there is no gradient explosion phenomenon. This conclusion can be reached by analyzing the parameter change curves that occur during the training process and displaying them in Fig.10 and Fig.11. The precision (P) and the recall (R) are getting closer to 1.

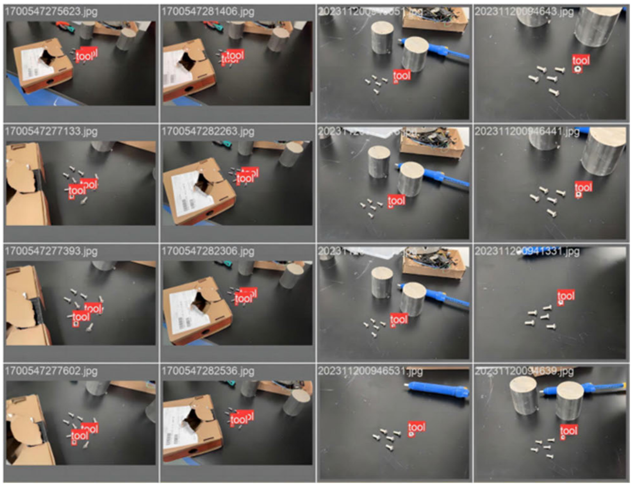


FIGURE 12. Performance of the model trained with synthetic samples.

Comparison of models are trained with synthetic samples and real samples on real datasets: For the purpose of training the detection model, the YOLOv8 model makes use of synthetic samples that are produced using an automatic annotation method that is created based on the Unity engine. After being trained with synthetic data, a target detection model is then used to recognize target objects in real experimental contexts. This method allows us to accurately evaluate the model's performance in detecting target objects in real-life scenarios. As can be seen in the left panel of Fig.12, the screws and nuts that are located on the experimental bench in the actual experimental environment are positioned at a variety of angles and in a variety of directions. All the target detection models that were trained with the synthesized samples are able to properly determine the precise location of the target object nut, and they are also able to construct bounding boxes for the discovered target object based on its location. Fig.12 illustrates that the YOLOv8 model employs a target detection model trained with the equal number of real and synthetic samples from the training set. Additionally, the position detection of the target object and the drawing of the bounding box in the model tests are performed using the same batch of test sets as the synthetic samples.

In order to investigate the performance of target detection models that have been trained using synthetic data in real detection situations in a manner that is both more thorough and intuitive, we not only investigated the ability of models trained with synthetic and real samples to recognize objects in real detecting contexts but also evaluated the performance of these models under varying illumination circumstances in the experiment. These results of detection parameters of models trained with an equal number of synthetic and actual samples

TABLE 2. Accuracy analysis of obtaining samples with equal quantities.

Training Set	Test Set	P	R	mAP
real samples	normal light	0.990	0.892	0.960
real samples	weak ligth	0.875	0.808	0.800
real samples	strong light	0.903	0.743	0.851
synthetic samples	normal light	0.934	0.876	0.941
synthetic samples	weak ligth	0.849	0.850	0.788
synthetic samples	strong ligth	0.872	0.850	0.808

TABLE 3. Accuracy analysis of obtaining samples with equal time.

Training Set	Number	P	R	mAP
real samples	200	0.959	0.822	0.880
synthetic samples	2000	0.981	0.906	0.965

on a batch of 100 real test sets are displayed in Table.2. The target detection model trained on a sample set consisting of 800 synthetic samples achieves precision (P) is 0.934, recall (R) is 0.876, mAP is 0.941 in a normal light detection environment. The target detection model trained on a sample set consisting of 800 real samples achieves P is 0.99, R is 0.892, and mAP is 0.96 in a normal light detection environment. The target detection model trained on synthetic samples achieved P is 0.849, R is 0.85, and mAP is 0.788 in a low-light detection environment. The target detection model trained on real samples achieved P is 0.875, R is 0.808, and mAP is 0.8 in the same low-light detection environment. The target detection model trained using synthetic samples achieved P is 0.872, R is 0.85, and mAP is 0.808 in a high-intensity light detection environment. Conversely, the target detection model trained using real samples achieve P is 0.903, R is 0.743, and mAP is 0.851 in the same light detection environment.

In the same time period, when manual annotation can get two hundred genuine samples that have been labeled, the sample fast annotation method can almost obtain two thousand synthetic samples that have been labeled using the same method. An actual test set is being used to test the identification of target detection models that have been trained on a variety of samples at the same time. The results of the tests are presented in Table.3, where the synthetic samples that were collected at the same time as the real samples are subjected to the identical test set. The precision (P) of the target detection model that was trained on a sample set that contained two hundred real samples is 0.959, the recall(R) is 0.822, and mAP is 0.88 when it was used in a real detection environment. The precision(P) of the target detection model that was trained using a sample set that had two thousand synthetic samples is 0.981, R is 0.906, and mAP is 0.965 when it was tested in a real detection scenario.

According to the results of the aforementioned experiments, the detection outcomes of the target detection model trained using synthetic samples and the target detection model trained using an equal number of real samples for real industrial part detection are as follows: mAP exhibited a disparity of 1.9% under normal lighting conditions and 1.2% under dimmer lighting conditions. The model trained on synthetic data collected outperforms the target detection model trained on real samples, resulting in an 8.5% increase in mAP for accurate detection. The comparison of the data indicates that the actual detection performance of the target detection model, which trained with synthetic samples, is essentially equivalent to the model trained with the equal number of actual industrial part samples. Under the assumption that the same amount of time is spent, the target detection model which trained with the synthetic samples that have been collected has a greater accuracy and a better detection effect than the model trained with the actual samples. This indicates that synthetic samples can take the place of genuine samples when it comes to carrying out activities related to industrial detection. In the trials described above, acquiring an equivalent quantity of synthetic samples requires significantly less expenditure and time compared to collect an equivalent quantity of real samples. It is noteworthy that there is a difference in the execution effect between the model trained with synthetic samples and the model trained with real samples when it comes to actual industrial detection. When there is a slight disparity between the simulated virtual detection environment and the actual detection environment, the model trained with synthetic samples and the model trained with real samples exhibit similar performance when conducting real detection. When spending the same time and cost, the model trained with synthetic samples performs better than the model trained with real samples. The experimental results demonstrate that the method proposed in this paper to produce synthetic samples is convincing and efficient in minimizing the costs associated with collecting samples during the target detection task. Different from other methods like GANs or Diffusion Models which generate datasets through manual labelling, the method proposed in this paper realistically replicates the actual detection environment to generate rich annotated datasets. Enterprises can control their data, create their own efficient models, reduce carbon dioxide emissions, save computational costs and time, and achieve sustainable business economics through the method to generate synthetic datasets proposed in this paper.

V. CONCLUSION

In this paper, we propose a method to generate synthetic datasets based on digital twins. The Unity engine is being introduced to imitate the actual detection environment by creating three-dimensional digital twins through modeling. The proposed method utilizes the features of the Unity engine in the virtual detection scenario to efficiently create annotated detection samples. The effectiveness of the synthetic sample set in the detection task is then demonstrated by comparing with the experimental data obtained from the actual dataset. The results of the experiments indicate that there will not be significant a distinction in accuracy between synthetic samples and real samples when it comes to performing the target detection task in actual detection environment. Furthermore, the cost of collecting synthetic samples is low which indicates that synthetic samples can be used in taking place of real samples to fulfillment the detection task in actual industrial inspection. When actual samples cannot be acquired because of privacy concerns or cannot be obtained due to safety concerns and accident risks in some specialized testing tasks in industrial areas, the significance of synthetic samples is particularly pronounced. During industrial production, enterprises often have trouble in obtaining a wide range of diverse samples due to limitations in their production processes. As a result, they are unable to collect a significant number of samples and the efficiency in acquiring samples is limited. Nevertheless, synthetic samples can manipulate the environmental conditions in the digital twins to produce diverse categories of samples. These synthetic samples can serve as a limited set of real samples to supplement the training data and enhance the model's performance. Additionally, synthetic samples can help reduce the expenses spending in acquiring real samples. Digital twins can replicate genuine inspection settings for virtual inspection scenarios. The proposed sample production method can rapidly produce numerous annotated synthetic samples. This efficient target detection system not only cuts down on the expenses and time associated with industrial inspection but also accelerates the execution of inspection tasks. Meanwhile, it addresses the issue of accuracy degradation in the model caused by imbalanced and insufficiently diverse samples in inspection tasks, resulting in improving benefits for the enterprises in terms of production execution.

This system also have a few restrictions within it. Difficulties in the modeling accuracy may arise when virtual scenarios attempt to replicate real scenarios, resulting in discrepancies between the model and the actual inspection scenario during the actual detection environment. In particular, the accuracy of the models designed for performing industrial inspection tasks is insufficient such as detecting defects in simulated samples. To enhance the resemblance between synthetic and real samples, we can explore the utilization of data augmentation techniques in conjunction with Generative Adversarial Networks (GANs) to augment the samples used in experiments in future research. Apart from this, we can consider employing more precise scanning equipment to model the virtual detection scene, ensuring a high level of similarity to the real scene. This approach will enable us to align the characteristics of the target objects in the synthetic samples with those in the real samples. Last but not least, while reviewing the effectiveness of a target detection system, it is possible to take into account a wider variety of count test set data for the purpose of testing model.

REFERENCES

[1] F. Liu, J. Tang, J. Yang, and H. Wang, "Automated industrial crack inspection system based on edge-edge collaboration of multiple cameras and programmable logic controller," in *Proc. IEEE Int. Symp. Broadband Multimedia Syst. Broadcast. (BMSB)*, Jun. 2023, pp. 1–4.

[2] C. Huo and T. Li, "Research on intelligent manufacturing equipment visual inspection control based on machine vision," in *Proc. IEEE 4th Int. Conf. Autom., Electron. Electr. Eng. (AUTEEE)*, Nov. 2021, pp. 374–379.

[3] L. BinYan, W. YanBo, C. ZhiHong, L. JiaYu, and L. JunQin, "Object detection and robotic sorting system in complex industrial environment," in *Proc. Chin. Autom. Congr. (CAC)*, Oct. 2017, pp. 7277–7281.

[4] Q. Lu and C. You, "Improved the detection algorithm of steel surface defects based on YOLOv7," in *Proc. 4th Int. Symp. Comput. Eng. Intell. Commun. (ISCEIC)*, Aug. 2023, pp. 104–107.

[5] M. Farag, A. N. A. Ghafar, and M. H. ALSIBAI, "Real-time robotic grasping and localization using deep learning-based object detection technique," in *Proc. IEEE Int. Conf. Autom. Control Intell. Syst. (I2CACIS)*, Jun. 2019, pp. 139–144.

[6] J. Terven and D. Cordova-Esparza, "A comprehensive review of YOLO architectures in computer vision: From YOLOv1 to YOLOv8 and YOLO-NAS," 2023, *arXiv:2304.00501*.

[7] Q. Tao, Y. Chen, and H. Chen, "A detection approach for wafer detect in industrial manufacturing based on YOLOv8," in *Proc. CAA Symp. Fault Detection, Supervision Saf. Tech. Processes (SAFEPROCESS)*, Sep. 2023, pp. 1–6.

[8] S. Jain, S. Dash, R. Deorari, and Kavita, "Object detection using COCO dataset," in *Proc. Int. Conf. Cyber Resilience (ICCR)*, Oct. 2022, pp. 1–4.

[9] P. Bergmann, M. Fauser, D. Sattlegger, and C. Steger, "MVTec AD—A comprehensive real-world dataset for unsupervised anomaly detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 9584–9592.

[10] R. Liu, Z. Yu, D. Mo, and Y. Cai, "An improved faster-RCNN algorithm for object detection in remote sensing images," in *Proc. 39th Chin. Control Conf. (CCC)*, Jul. 2020, pp. 7188–7192.

[11] B. Drost, M. Ulrich, P. Bergmann, P. Hartinger, and C. Steger, "Introducing MVTec ITODD—A dataset for 3D object recognition in industry," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops (ICCVW)*, Oct. 2017, pp. 2200–2208.

[12] C. S. Islam and M. Alauddin, "A novel idea of classification of E-commerce products using deep convolutional neural network," in *Proc. 4th Int. Conf. Electr. Eng. Inf. Commun. Technol. (iCEEiCT)*, Sep. 2018, pp. 342–347.

[13] M. Zhao, Y. Yang, K. Liu, D. Yan, and Z. Liu, "A object detection model with multiple data enhancements," in *Proc. Int. Conf. Artif. Intell., Inf. Process. Cloud Comput. (AIIPCC)*, Aug. 2022, pp. 161–164.

[14] V. Gautam, R. G. Tiwari, A. Misra, D. Witarsyah, N. K. Trivedi, and A. K. Jain, "Dry fruit classification using deep convolutional neural network trained with transfer learning," in *Proc. Int. Conf. Advancement Data Sci., E-Learn. Inf. Syst. (ICADEIS)*, Aug. 2023, pp. 1–6.

[15] K. Baek and H. Shim, "Commonality in natural images rescues GANs: Pretraining GANs with generic and privacy-free synthetic data," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 7844–7854.

[16] F. Gao, Q. Liu, J. Sun, A. Hussain, and H. Zhou, "Integrated GANs: Semi-supervised SAR target recognition," *IEEE Access*, vol. 7, pp. 113999–114013, 2019.

[17] F.-A. Croitoru, V. Hondru, R. T. Ionescu, and M. Shah, "Diffusion models in vision: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 9, pp. 10850–10869, Sep. 2023.

[18] C. Pang and Y. Cheng, "Detection of river floating waste based on decoupled diffusion model," in *Proc. 8th Int. Conf. Autom., Control Robot. Eng. (CACRE)*, Jul. 2023, pp. 57–61.

[19] M. Shafto, M. Conroy, R. Doyle, E. Glaessgen, C. Kemp, J. LeMoigne, and L. Wang, *Modeling, Simulation, Information Technology and Processing*. Washington, DC, USA: National Academies Press, 2012, pp. 282–293.

[20] S. Wang and Y. Zhao, "Image recognition technology and digital twin in the power industry," in *Proc. IEEE 2nd Int. Conf. Electr. Eng., Big Data Algorithms (EEBDA)*, Feb. 2023, pp. 1398–1401.

[21] N. Napp, "Digital twins in the real world," in *Proc. IEEE 2nd Int. Conf. Intell. Reality (ICIR)*, Dec. 2022, pp. 17–20.

[22] Z. Min, S. Zhou, Z. Kang, S. Shekhar, C. Mahmoudi, S. Gokhale, and A. Gokhale, "Managing and optimizing 5G & beyond network resources for multi-task digital twin applications in Industry 4.0," in *Proc. IEEE 26th Int. Symp. Real-Time Distrib. Comput. (ISORC)*, May 2023, pp. 220–223.

[23] Y. Liu, K. Zhang, and Z. Li, "Application of digital twin and parallel system in automated driving testing," in *Proc. IEEE 1st Int. Conf. Digit. Twins Parallel Intell. (DTPI)*, Jul. 2021, pp. 123–126.

[24] G. Poechgraber, S. Bougain, B. Wallner, G. Bohaty, T. Trautner, and F. Bleicher, "Introduction of a digital twin for the product development phase," in *Proc. Int. Conf. Eng. Manage. Commun. Technol. (EMCTECH)*, Oct. 2023, pp. 1–6.

[25] M. Shrivastava, R. Chugh, S. Gochhait, and A. B. Jibril, "A review on digital twin technology in healthcare," in *Proc. Int. Conf. Innov. Data Commun. Technol. Appl. (ICIDCA)*, Mar. 2023, pp. 741–745.

[26] D. Baimukashev, A. Zhilisbayev, A. Kuzdeuov, A. Oleinikov, D. Fadeyev, Z. Makhataeva, and H. A. Varol, "Deep learning based object recognition using physically-realistic synthetic depth scenes," *Mach. Learn. Knowl. Extraction*, vol. 1, no. 3, pp. 883–903, Aug. 2019.

[27] A. Damian, C. Filip, A. Nistor, I. Petrariu, C. Mariuc, and V. Stratan, "Experimental results on synthetic data generation in unreal engine 5 for real-world object detection," in *Proc. 17th Int. Conf. Eng. Modern Electric Syst. (EMES)*, Jun. 2023, pp. 1–4.

[28] S. Borkman, A. Crespi, S. Dhakad, S. Ganguly, J. Hogins, Y.-C. Jhang, M. Kamalzadeh, B. Li, S. Leal, P. Parisi, C. Romero, W. Smith, A. Thaman, S. Warren, and N. Yadav, "Unity perception: Generate synthetic data for computer vision," 2021, *arXiv:2107.04259*.

[29] K. Wang, K. Adimulam, and T. Kesavadas, "Tetrahedral mesh visualization in a game engine," in *Proc. IEEE Conf. Virtual Reality 3D User Interface (VR)*, Mar. 2018, pp. 719–720.

[30] A. I. Károly, Á. Károly, and P. Galambos, "Automatic generation and annotation of object segmentation datasets using robotic arm," in *Proc. IEEE 10th Jubilee Int. Conf. Comput. Cybern. Cyber-Med. Syst. (ICCC)*, Jul. 2022, pp. 63–68.

ZHE DONG received the B.Sc. degree in engineering from Beihang University, in 2004, and the Ph.D. degree in engineering from the Institute of Automation, Chinese Academy of Sciences, in 2009. Since 2009, he has been teaching with the Department of Automation, North China University of Technology, Beijing, China. From 2014 to 2015, he was a Visiting Scholar with the University of Michigan. He is currently a Professor and the Doctoral Director with the North China University of Technology. He is also the Dean of the School of Electrical and Control Engineering, North China University of Technology, and the Director of Beijing Key Laboratory of Fieldbus Technology and Automation. His research interests include information detection and intelligent processing, networked control systems, industrial internet, and environmental testing and control.

YUE YANG is currently pursuing the Master of Engineering degree with the North China University of Technology, Beijing, China. Her research interests include artificial intelligence, image recognition, and machine vision.

ANQI WANG received the Ph.D. degree in instrument science and technology from Beihang University. She is currently a Lecturer with the North China University of Technology. Her research interests include wind turbine intelligent operation and maintenance, and power prediction.

TIANXU WU is currently pursuing the Master of Engineering degree with the North China University of Technology, Beijing, China. His current research interest includes neural networks.