mathematics

MDPI

Article

Synthetic Data Generation Based on RDB-CycleGAN for Industrial Object Detection

Jiwei Hu, Feng Xiao, Qiwen Jin *, Guangpeng Zhao and Ping Lou

School of Information Engineering, Wuhan University of Technology, Wuhan 430070, China; hujiwei@whut.edu.cn (J.H.); 320668@whut.edu.cn (F.X.); zhaoguangpeng@whut.edu.cn (G.Z.); louping@whut.edu.cn (P.L.)

* Correspondence: qiwenjin@whut.edu.cn

Abstract: Deep learning-based methods have demonstrated remarkable success in object detection tasks when abundant training data are available. However, in the industrial domain, acquiring a sufficient amount of training data has been a challenge. Currently, many synthetic datasets are created using 3D modeling software, which can simulate real-world scenarios and objects but often cannot achieve complete accuracy and realism. In this paper, we propose a synthetic data generation framework for industrial object detection tasks based on image-to-image translation. To address the issue of low image quality that can arise during the image translation process, we have replaced the original feature extraction module with the Residual Dense Block (RDB) module. We employ the RDB-CycleGAN network to transform CAD models into realistic images. Additionally, we have introduced the SSIM loss function to strengthen the network constraints of the generator and conducted a quantitative analysis of the improved RDB-CycleGAN-generated synthetic data. To evaluate the effectiveness of our proposed method, the synthetic data we generate effectively enhance the performance of object detection algorithms on real images. Compared to using CAD models directly, synthetic data adapt better to real-world scenarios and improve the model's generalization ability.

Keywords: synthetic data; RDB-CycleGAN; image translation; object detection

MSC: 68T07; 68U10

Citation: Hu, J.; Xiao, F.; Jin, Q.; Zhao, G.; Lou, P. Synthetic Data Generation Based on RDB-CycleGAN for Industrial Object Detection. *Mathematics* **2023**, *11*, 4588. https://doi.org/10.3390/math11224588

Academic Editor: Jonathan Blackledge

Received: 10 October 2023
Revised: 30 October 2023
Accepted: 3 November 2023
Published: 9 November 2023

1. Introduction

The advent of the Industry 4.0 era has brought enormous opportunities and challenges to the industrial sector [1]. In this digital and intelligent age, object detection technology has become particularly crucial in the industrial domain. As one of the key technologies in industrial automation and intelligent manufacturing, object detection provides strong support for the intelligent perception of industrial equipment, automated control of production processes, and improvement of product quality [2].

However, in industrial object detection, we face a series of challenges. Firstly, industrial environments often exhibit complex and dynamic characteristics, involving a wide variety of object types and shapes [3]: for instance, various components on a production line, different types of packaging, and more. These objects may come in different sizes, shapes, colors, and materials, and they might overlap, occlude, or be positioned at various orientations and angles. This diversity makes it difficult for object detection algorithms to adapt to such highly varied situations, leading to insufficient detection accuracy and robustness [4].

Secondly, data collection and annotation in industrial production processes are typically labor-intensive and time-consuming tasks. Data annotation requires skilled personnel to manually label objects, such as bounding boxes or assigning labels to them. This process

consumes a lot of human resources and time, and usually requires professional knowledge. Particularly for certain industrial scenarios or rare industrial components, there may be no readily available datasets to use. This results in a relative scarcity of datasets that can be used for training, limiting the performance of object detection algorithms in industrial applications [5].

To address the issue of the lack of datasets in industrial object detection, we have noticed many methods of data augmentation, such as the Object-Based Augmentation approach proposed by Svetlana Illarionova et al. in the field of remote sensing [6]. Additionally, Golnaz Ghiasi et al. proposed a simple copy-and-paste method for data augmentation [7]. Moreover, researchers began to use synthetic datasets [8–10]. Synthetic datasets are created using computer graphics techniques and simulated physical processes to generate realistic synthetic images with corresponding annotated information of the targets. Compared to real datasets, synthetic datasets offer advantages such as rapid acquisition, flexible generation, and customization for different scenarios and tasks. B. Kiefer et al. utilized synthetic data for Unmanned Aerial Vehicle (UAV) object detection [11]. Through synthetic datasets, we can create more complex and diverse industrial scenes without the limitations of actual data collection. This helps expand the training data and improves the adaptability and generalization of object detection algorithms in industrial environments.

However, when using synthetic datasets, we also need to carefully consider their realism and effectiveness. Since synthetic data are generated by models, there may be some differences compared to real data [12]. Farzan Erlik Nowruzi et al. analyzed object detection performance using synthetic and real data [13]. Therefore, we need to adopt a series of methods to ensure the consistency between synthetic data and real data in terms of feature distribution and data distribution, thus ensuring the accuracy and robustness of the model in real industrial scenarios.

To overcome the differences between synthetic and real data and achieve image translation from the synthetic image domain to the real image domain, researchers can leverage Generative Adversarial Networks (GANs) [14], a powerful deep learning tool [15]. GANs consist of a generator and a discriminator, forming an adversarial model. The generator is responsible for generating synthetic data, while the discriminator is tasked with distinguishing between real data and the synthetic data generated by the generator. Both components continuously optimize their performance through adversarial training. The generator aims to generate increasingly realistic synthetic data, while the discriminator aims to accurately determine whether the input data are real or synthetic [16].

In recent years, the image translation technology of GANs has made significant progress and has been widely applied in various fields [17,18]. In industrial object detection, using GANs to generate realistic synthetic data not only increases the diversity and quantity of training data but also improves the adaptability of the object detection model to the complexities and variations in real industrial scenarios [19]. However, ensuring that the synthetic data generated by GANs matches the quality and diversity of real data still requires careful design and tuning. Furthermore, to maintain the good generalization capabilities of the object detection algorithm after training with synthetic data, it is essential to strike a moderate balance between synthetic and real data during the training process. Combining other data augmentation techniques is also necessary to enhance the diversity of the data.

In the task of image translation [20], we can regard the generator as a converter from the synthetic image domain to the real image domain. Through training GANs, the generator learns to transform synthetic images into images that resemble real data, thereby reducing the differences between synthetic and real data. This allows us to augment the real dataset with synthetic data, thus improving the performance of object detection algorithms in real industrial environments.

In this paper, we introduce a novel approach to provide more training data for object detection tasks in the industrial domain. Our main contributions are as follows:

1. We propose a synthetic data generation framework for industrial object detection tasks, enabling the effortless creation of a larger volume of industrial part data using a small number of real industrial part images and CAD models.
2. To enhance the quality of generated images in achieving the transformation task from CAD models to real images, we have replaced the original feature extraction module with an RDB (Residual Dense Block) module. Additionally, we have introduced an SSIM (Structural Similarity Index Measure) loss function to strengthen the network constraints of the generator. The real images obtained through the RDB-CycleGAN network contribute to augmenting our dataset.
3. Experiments show that the synthetic data obtained through our method has a significant competitive advantage, effectively augmenting industrial part data and partially bridging the gap between synthetic and real data.

2. Related Work

To address the issue of limited datasets in industrial object detection, researchers have started exploring the use of synthetic datasets. In this section, we primarily focus on methods related to synthetic data generation for object detection and image translation networks.

2.1. Overview of Object Detection

Object detection in the industrial domain has been one of the hot research topics in recent years, as it is of significant importance in improving production efficiency, ensuring product quality, and achieving industrial intelligence. Many researchers have proposed various object detection algorithms and solutions for different industrial scenarios and tasks. In traditional industrial object detection, researchers often rely on manually designed feature extraction and detection algorithms. For example, methods based on features like Histogram of Oriented Gradients (HOG) and Scale-Invariant Feature Transform (SIFT) have been widely used in industrial object detection tasks over the past decade [21]. However, these methods often require a large amount of manual labor and expertise and have limited detection performance in complex scenes. With the rise of deep learning, Convolutional Neural Networks (CNNs) have made significant progress in industrial object detection. CNNs, through end-to-end learning, can automatically learn more efficient and useful feature representations from data, leading to excellent performance in complex scenes for object detection algorithms. Among them, Faster R-CNN, YOLO, and SSD [22,23] have become representative methods in the field of industrial object detection. Faster R-CNN introduces a Region Proposal Network (RPN) to optimize the process of candidate box generation for objects; YOLO (You Only Look Once) adopts a one-stage detection approach, achieving a good balance between high detection speed and accuracy; SSD (Single Shot Multibox Detector) combines multi-scale features for object detection, improving the detection capability for small objects. Despite the significant progress of deep learning methods in industrial object detection, challenges remain in practical applications, such as data scarcity and adaptability to complex industrial environments. To address these issues, some researchers have started exploring the use of synthetic datasets to increase training data [24]. Synthetic datasets, generated using techniques like Generative Adversarial Networks (GANs), can simulate data from real scenarios, thereby enhancing the diversity and quantity of training data. This approach provides new perspectives for industrial object detection and opens up new possibilities for efficient object detection in complex industrial environments [25].

2.2. Synthetic Data Generation

Synthetic data have been extensively researched and applied in the fields of computer vision and machine learning. With the advancement of deep learning techniques, synthetic data have become an effective approach to address data scarcity and generalization issues [26]. In the domain of industrial object detection, the use of synthetic data has also gained attention among researchers. Synthetic data are typically generated using Genera-

tive Adversarial Networks (GANs) or other generative models [27]. GANs can generate synthetic data that closely resembles real data through an adversarial process of training the generator and discriminator. The generator aims to produce realistic synthetic data, while the discriminator strives to differentiate between real and synthetic data. As the training progresses, the generator continuously improves, and the generated synthetic data become increasingly similar to the distribution of real data. The advantages of using synthetic data in industrial object detection lie in the ability to rapidly obtain large quantities of diverse data, especially when real data are difficult to obtain or costly [28]. Synthetic datasets can flexibly generate different types and shapes of target objects based on the needs of various industrial scenarios and tasks. Additionally, synthetic data allow for control over factors such as lighting, angles, and backgrounds, thereby enhancing the robustness and generalization capabilities of object detection algorithms. Some studies have shown that joint training with synthetic and real data can significantly improve model performance in object detection tasks. For instance, using synthetic data as an auxiliary dataset and employing transfer learning via pretraining the model on synthetic data and fine-tuning it on real data can enhance the model's performance on real datasets [29]. Furthermore, using synthetic data for data augmentation can increase the diversity of training data, thereby improving the model's adaptability to complex scenes.

However, the use of synthetic data also poses some challenges. Firstly, the generated synthetic data need to exhibit certain consistency with real data in terms of feature and data distributions; otherwise, the model's performance in real scenarios may decline [30]. Secondly, the quality of synthetic data significantly impacts the final model's performance. Ensuring that the generated synthetic data are sufficiently realistic is a crucial concern. Therefore, when utilizing synthetic data, careful design of synthetic data generation strategies, along with the incorporation of other data augmentation techniques, is necessary to ensure that the generated synthetic data positively contribute to the training of object detection algorithms [31].

2.3. The CycleGAN-Based Image Translation Networks

To address the issue of data scarcity in industrial object detection, some researchers have begun to explore the use of synthetic data and utilize a variant model of Generative Adversarial Networks (GANs) called CycleGAN to achieve image translation from the CAD image domain to the real image domain [32,33]. CycleGAN, proposed by Zhu et al. in 2017 [34], stands out for its ability to perform unpaired image translation, enabling bidirectional image conversion between two different domains while maintaining content consistency. In industrial object detection, CAD images are typically used by engineers for design and simulation, while real images are collected during the industrial production process. There is a significant difference between these two image domains, and traditional data augmentation and transfer learning methods often yield limited results in this scenario. Therefore, using CycleGAN for image translation has emerged as a novel solution. Through CycleGAN, CAD images can be translated into real images, thereby generating more realistic and diverse data in an industrial environment. This approach helps alleviate the data scarcity problem in industrial object detection and improves the model's generalization capabilities in real scenes. Additionally, CycleGAN can also perform inverse translation from the real image domain to the CAD image domain, converting real images into CAD images, further enriching the diversity of synthetic data.

However, the application of CycleGAN also faces challenges [35,36]. Firstly, the generated synthetic images need to possess sufficient realism and credibility to ensure the performance of the object detection model in real scenes. Therefore, careful parameter tuning and optimization of CycleGAN are required to obtain high-quality synthetic images. Secondly, there may be significant differences between CAD images and real images, necessitating the reasonable design of loss functions and weights during the training process to allow CycleGAN to learn better image conversion mappings [37–39].

3. Proposed Method

As illustrated in Figure 1, the proposed method mainly contains two steps: (1) Training images are collected for the image-to-image translation model, including industrial part CAD models and real part images from the scene. We cropped the CAD images and real images to obtain images of size 256 × 256. These collected images are cropped and preprocessed to obtain source domain X images and target domain Y images for the image-to-image translation model. (2) The preprocessed training images are fed into the image-to-image translation model based on unpaired GAN, where the model learns the detailed characteristics of the parts from real images to achieve the image translation from CAD models to real images.

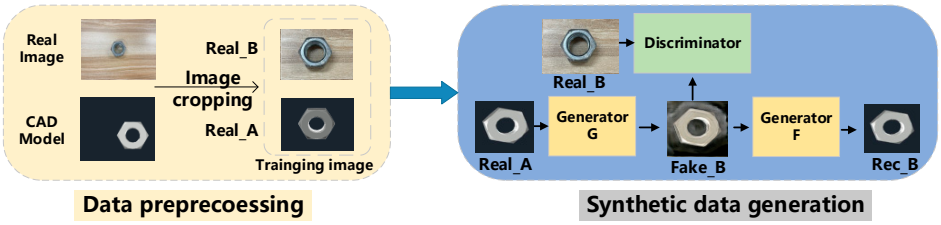


Figure 1. Synthetic data generation for object detection.

3.1. Model Architecture

In this paper, we use a model based on the CycleGAN network structure for the task of image-to-image translation. The model consists of two generators and two discriminators, performing the conversion between CAD model images and real scene images in an unpaired training dataset to generate synthetic data. The overall model architecture is illustrated in Figure 2. This model forms a circular network structure composed of two mirrored Generative Adversarial Networks (GANs), comprising two generators and two discriminators. The loss functions include adversarial loss, cycle consistency loss, and structural similarity loss. In the forward network, the generator G maps data Real_A from source domain X to target domain Y, producing Fake_B. The discriminator makes judgments on the generated image Fake_B, and then the generator F reconstructs it back to data Rec_A in domain X. Similarly, the transformation from the target domain Y to the source domain X follows the same process. The network can learn the mapping between the source domain and the target domain, and it can also reconstruct back from the target domain, achieving the task of transforming CAD models into real part images.

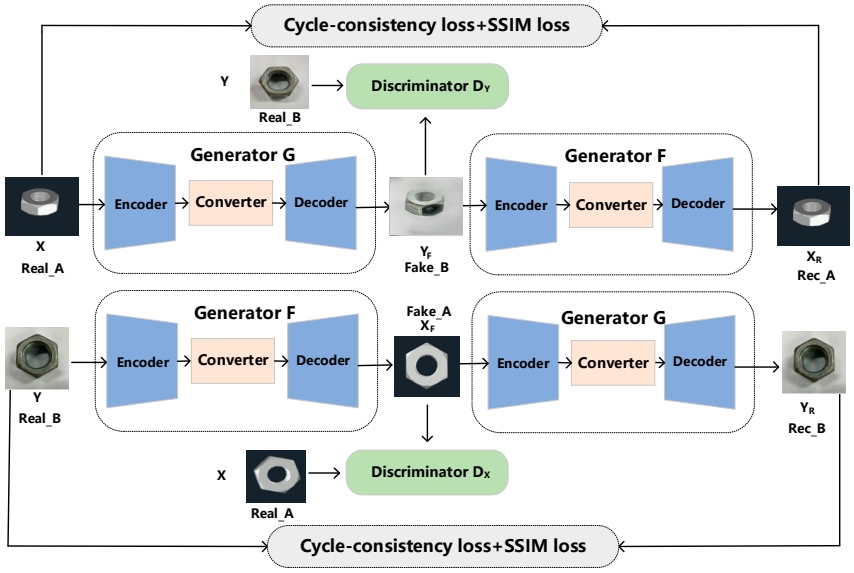


Figure 2. Structure of RDB-CycleGAN model.

During the model training process, first, for the input image domains X and Y, the Generative Adversarial Networks generate corresponding fake and reconstructed images. Then, the gradients of the generator network are computed, and the weights of the generator network are updated accordingly; next, the gradients of the discriminator network are calculated, and the weight coefficients of the discriminator network are updated. Finally, the latest network model is saved based on the set frequency parameter for model saving. The pseudo-code of the model algorithm is shown in Algorithm 1:

Algorithm 1: RDB-CycleGAN image translation algorithm

Input: image domain ***X***, image domain ***Y***, model training epoch ***N***, ***iters***, ***save_model _freq***
Output: ***model***

for each epoch in (1,*N*) **do**
  for each data in dataset **do**
    Generate domain *X* image fake_x and domain *Y* image fake_y;
    Set the gradient of the generated networks *G* and *F* to 0;
    Calculate the gradient of the generated network *G* and *F*;
    Update the weight parameters of the generated networks *G* and *F*;
    Set the gradient of D_X and D_Y to 0 for the discriminant network;
    Calculate the gradient of the discriminant network D_X and D_Y;
    Update the weight parameters of D_X and D_Y discriminant networks;
  end for
  if iters % sava_model_freq == 0
    Save the latest model
  end if
end for

3.2. Network Structure

The CycleGAN network has a mirrored structure and consists of two parts, each of which is a sub-network based on GAN. The generator of each GAN network is composed of an encoder, a transformation module, and a decoder. The encoder module consists of two convolutional layers with a stride of 2 and a kernel size of 3 × 3. When the input image size is 128 × 128, the transformation module consists of 6 residual blocks with a kernel size of 3 × 3, and when the input image is 256 × 256, it consists of 9 residual blocks with a kernel size of 3 × 3. The decoder module consists of two transposed convolutional layers with a kernel size of 3 × 3, and the modules are connected through a fully convolutional network.

To achieve the image translation task from industrial part CAD images to real images and improve the network performance, we have adopted a series of improvement measures. Firstly, we optimized the generator of the CycleGAN network through replacing the original 9 ResNet blocks with 5 newly designed Dense Blocks, which we call RDB (Residual Dense Block) modules. Each Dense Block module consists of 4 Conv + LReLU structures and one convolutional layer, and these layers are densely connected, enabling the network to extract deeper-level feature information. Compared to the traditional ResNet structure, the advantage of RDB modules lies in their increased network depth, allowing us to capture complex image features more effectively, thus enhancing translation accuracy and stability. In addition, to further improve network performance, we used Conv + LReLU structures in the Dense Blocks instead of the Conv + BN + ReLU structures used in ResNet, and we removed batch normalization. This change resulted in significant performance improvement, not only increasing network stability but also reducing artifacts in the generated images, leading to a notable enhancement in the overall image quality. The RDB module and the improved generator structure are shown in Figures 3 and 4. In Figure 3, the dashed lines indicate the structure of a Dense Block. The yellow lines, blue lines, green lines, and brown lines represent dense connection layers.

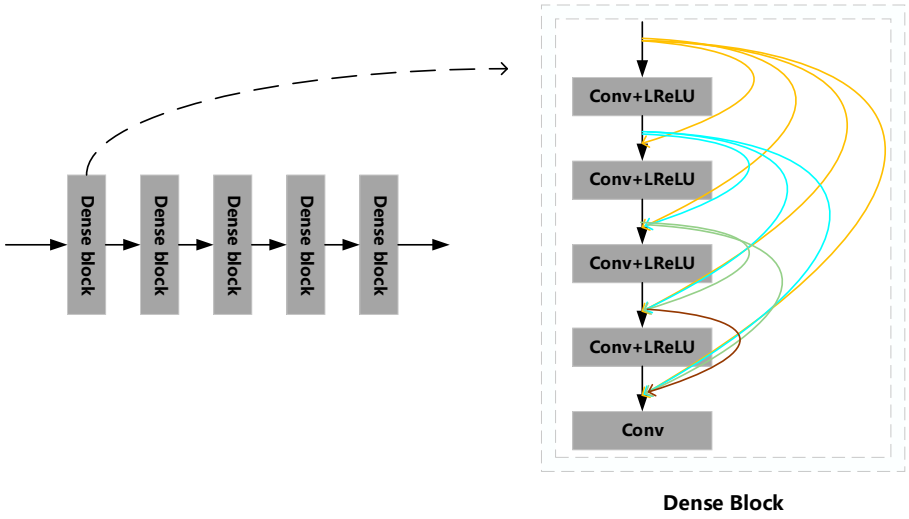


Figure 3. The RDB module and Dense Block of the RDB-CycleGAN network.

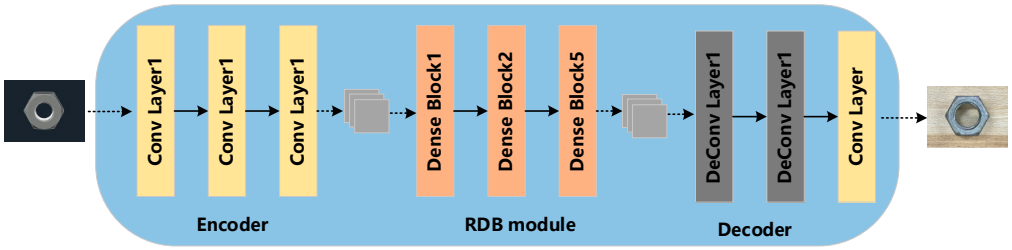


Figure 4. The architecture of the generator network.

The discriminator of the GAN network in this paper adopts the original network's 70 × 70 PatchGAN network structure. Compared to traditional generative adversarial networks, this structure can better capture local features in the images. PatchGAN maps the input feature map into a 30 × 30-sized output feature map, which corresponds to the probabilities of multiple 70 × 70 local patches of the input feature map being real. The discriminator convolves over the entire N × N-sized image, resulting in a 30 × 30-sized output, and then takes the average value to obtain the final output. The structure of the discriminator is shown in Figure 5.

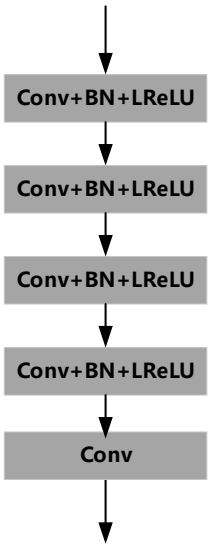


Figure 5. The architecture of the discriminator network.

3.3. Loss Function

CycleGAN uses cycle consistency to establish the mapping relationship between the source domain and the target domain. The model employs both adversarial loss and cycle consistency loss in both Generative Adversarial Networks. Given two image domains X and Y, two mapping functions G: X→Y and F: Y→X are established between the two domains, where G and F are the generators in the GAN. With this, a GAN loss can be defined, and the adversarial loss from X to Y is as follows:

$$L_{GAN}(G, D_Y, X, Y) = E_{y \sim p_{data}(y)}[\log D_Y(y)] + E_{x \sim p_{data}(x)}[\log(1 - D_Y(G(x)))] \quad (1)$$

Similarly, the adversarial loss from Y to X is as follows:

$$L_{GAN}(F, D_X, Y, X) = E_{x \sim p_{data}(x)}[\log D_X(x)] + E_{y \sim p_{data}(y)}[\log(1 - D_X(F(y)))] \quad (2)$$

In this context, we denote X and Y as the source domain and target domain, respectively. $x \in$X, $y \in$Y. $p_{data}(x)$ represents the data distribution of the source domain X, and $p_{data}(y)$ represents the data distribution of the target domain Y. $E_{y \sim p_{data}(y)}$ indicates the expectation of y under the distribution $p_{data}(y)$, and $E_{x \sim p_{data}(x)}$ signifies the expectation of x under the distribution $p_{data}(x)$.

When learning the mappings from X to Y and Y to X, the image domain X is transformed by the generator G to generate the forged domain Y_F, and then through F to generate the reconstructed domain X_R. The goal is to minimize the difference between X and X_R through calculating their loss. Similarly, the aim is to minimize the difference between Y and Y_R, as much as possible. Therefore, the cycle consistency loss function is defined as follows:

$$L_{cyc}(G, F) = E_{x \sim p_{data}(x)}[\| F(G(x)) - x \|_1] + E_{y \sim p_{data}(y)}[\| G(F(y)) - y \|_1] \quad (3)$$

In addition, the Structural Similarity (SSIM) loss function is introduced, which calculates the structural similarity between the generated images and their corresponding real images. Gwantae Kim et al. utilized SSIM to enhance the quality of super-resolution images [40]. Similarly, Fengquan Zhang et al. employed SSIM in the context of improving the quality of image translation networks [41]. Through minimizing the SSIM loss, we encourage the generator to preserve more image structural information during the translation process, thereby enhancing the quality and realism of the generated images. The introduction of SSIM loss also helps to reduce artifacts and blurriness that may occur in the generated images, thereby improving the stability and reliability of the image translation. This metric defines the structural information of an image from the perspective of image composition, including luminance, contrast, and structure, reflecting the attributes of objects in the scene. Thus, SSIM models distortion as a combination of these three different factors: luminance, contrast, and structure. In image processing, the estimate of luminance information is represented by the mean, contrast information is represented by the standard deviation, and the degree of structural similarity is represented by the covariance. μ_x and μ_y are the mean values of pixels in domains X and Y, respectively, σ_x^2 and σ_y^2 are the variances of pixels in domains X and Y, and σ_{xy} is the covariance between domains X and Y. The formula for SSIM is as follows:

$$\mu_x = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} X_{i,j}, \quad (4)$$

$$\sigma_x^2 = \frac{1}{H \times W - 1} \sum_{i=1}^{H} \sum_{j=1}^{W} \Big(X_{i,j} - \mu_x \Big)^2, \quad (5)$$

$$\sigma_{xy} = \frac{1}{H \times W - 1} \sum_{i=1}^{H} \sum_{j=1}^{W} (Y_{i,j} - \mu_y)(X_{i,j} - \mu_x). \quad (6)$$

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{\left(\mu_x^2 + \mu_y^2 + c_1\right)\left(\sigma_x^2 + \sigma_y^2 + c_2\right)} \tag{7}$$

Using the above formula, we obtain the term for the Structural Similarity (SSIM) loss function, which is as follows:

$$L_{SSIM}(G,F) = [1 - SSIM(x, F(G(x)))] + [1 - SSIM(y, G(F(y)))] \tag{8}$$

The overall loss function of the RDB-CycleGAN network is a weighted combination of three parts: adversarial loss, cycle consistency loss, and structural similarity loss:

$$L(G,F,D_X,D_Y) = L_{GAN}(G,D_Y,X,Y) + L_{GAN}(F,D_X,Y,X) + \lambda L_{cyc}(G,F) + \theta L_{SSIM}(G,F) \tag{9}$$

Here, we set λ to 1 and θ to 0.2.

4. Experiments and Discussion

This paper conducted two main sets of experiments. Firstly, a comparison was made between the improved RDB-CycleGAN network and other image translation networks. The experimental results demonstrated the effectiveness of the proposed enhancement in improving the quality of image generation. Additionally, we applied the images generated by the RDB-CycleGAN network to the YOLOv5 object detection algorithm to demonstrate the effectiveness of synthetic data.

4.1. Experimental Detail

In the image translation task, CycleGAN [34], DualGAN [42], GP-UNIT [43], and StarGAN-v2 [44] have all demonstrated excellent performance. Therefore, we chose these algorithms for comparison with our RDB-CycleGAN network.

The computer configuration used in our experiments consists of an Nvidia GeForce RTX 2080Ti GPU and an Intel i9-13900K CPU. Regarding parameter settings, we set the epoch to 200, learning rate to 0.0002, and batch size to 8. During the network training process, we selected 1000 real images and 1000 CAD images for each of the three industrial parts to achieve image domain conversion.

We conducted comprehensive experiments and evaluations on the generated images to objectively assess their performance and quality. For this purpose, we employed several evaluation metrics, including Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index (SSIM), and Fréchet Inception Distance (FID). PSNR is a traditional image quality metric used to compare the distortion between original images and GAN-generated images. A higher PSNR value indicates less distortion and higher image quality between the generated and real images. SSIM is another widely used image similarity metric that considers both structural information and luminance contrast. A higher SSIM value closer to 1 indicates higher structural similarity between the generated and real images. FID is a metric used to assess the difference between two image distributions. It quantifies the difference between real and generated images through comparing their distributions in the feature space of an Inception network. A lower FID value indicates less difference between the generated and real images, indicating better performance of the generation model.

In our experiments, we calculated PSNR, SSIM, and FID to assess the quality of images generated by the adversarial network, using these metrics for a holistic performance evaluation. This approach offered an objective analysis, guiding enhancements in the network's generative efficiency. The FID score measures the distance between real and generated images at the feature level. Through using InceptionV3, we generate N × 2048 vectors for N images in the real dataset to obtain the mean μ_x. Similarly, for M images generated in the synthetic dataset, we generate M × 2048 vectors and obtain the mean μ_y. Σ_x represents the covariance matrix for the real dataset, Σ_y represents the generated dataset,

and T_r indicates the sum of the diagonal elements of the matrices. The FID is calculated using the following formula:

$$FID = || \mu_x - \mu_y ||^2 + T_r\left(\Sigma_x + \Sigma_y - 2(\Sigma_x\Sigma_y)^{1/2}\right). \quad (10)$$

Peak Signal-to-Noise Ratio (PSNR) is used to compare the similarity between two images and evaluate the level of distortion between compressed or processed images and the original image. Mean Squared Error (MSE) represents the mean squared difference between the two images, and MAX_I^2 denotes the maximum possible pixel value of the image. The calculation formula for PSNR is as follows:

$$PSNR = 10 \cdot log_{10}\left(\frac{MAX_I^2}{MSE}\right) \quad (11)$$

4.2. Experiment Results

We conducted image translation experiments using DualGAN, CycleGAN, GP-UNIT, StarGAN-v2, and the improved CycleGAN proposed in this paper. The specific experimental results are shown in Figure 6. In Figure 6, the initial column represents CAD images, while the second, third, fourth, and fifth columns display the corresponding experimental outcomes of image translation employing the DualGAN model, CycleGAN model, GP-UNIT model, StarGAN-v2, and the enhanced CycleGAN model, respectively.

From a subjective visual perspective, when using the original CycleGAN, DualGAN, GP-UNIT, and StarGAN-v2 networks for image style transfer, there are clearly many issues, such as significant structural deficiencies and excessive artifacts or incomplete images, which have a significant impact on subsequent object detection tasks. In comparison, the RDB-CycleGAN generates fewer artifacts and has minimal structural deficiencies, resulting in less object deformation.

Additionally, we randomly selected 200 images from each of the three categories of generated datasets for the objective evaluation of image quality. Table 1 demonstrates the quantitative results through the objective evaluation metrics. The best and second-best results are in red and blue, respectively. From Table 1, it can be observed that the RDB-CycleGAN algorithm performs better than the CycleGAN, DualGAN, GP-UNIT, and StarGAN-v2 algorithms in terms of SSIM, FID, and PSNR metrics for the style transfer from CAD models to real images. The generated images have better quality and are closer to the standard real images. The objective evaluation results of the metrics are consistent with the subjective visual perception.

Table 1. The quantitative comparison results of SSIM, FID, and PSNR.

Method	SSIM	FID	PSNR/dB
DualGAN	0.527	147.49	26.47
CycleGAN	0.643	**130.68**	28.15
GP-UNIT	0.619	138.27	**28.69**
StarGAN-v2	**0.655**	133.73	27.95
Ours	**0.684**	**124.64**	**29.74**

We conducted ablation experiments on the introduced RDB module and SSIM loss, and Figure 7 presents the results of our ablation experiments.

From a subjective visual perspective, replacing the generator in the CycleGAN model with the RDB module has effectively improved the part completion in the image translation process. The generated features now align better with the original images, resulting in clearer images. Moreover, the inclusion of the SSIM loss function in the network model has also led to a certain degree of enhancement in the part's structural features. Through

our ablation experiments, it is evident that our RDB module and SSIM loss function can effectively enhance the quality of the synthesized images.

Figure 6. Results of the different algorithms: (**a**) CAD image; (**b**) DualGAN; (**c**) CycleGAN; (**d**) GP-UNIT; (**e**) StarGAN-v2; (**f**) ours.

In addition, we employed two different data augmentation methods, CAD images and RDB-CycleGAN synthetic images, to explore their effectiveness in the object detection task. We used Yolov5 as the object detection model and conducted comparative experiments to evaluate the effects of these two data augmentation methods. For CAD images, we generated diverse images using computer-aided design techniques to simulate different perspectives in real-world scenarios. For the RDB-CycleGAN synthetic images, we employed image style transfer techniques to generate synthetic data, thereby increasing the

diversity and complexity of the dataset. We retrained the Yolov5 object detection model using the augmented datasets and evaluated its performance on the same test set.

Figure 7. Ablation study: (**a**) CAD model; (**b**) CycleGAN; (**c**) use RDB module; (**d**) add SSIM loss; (**e**) ours.

The YOLOv5 framework provides models of various sizes (Yolov5s, Yolov5m, Yolov5l, and Yolov5x) to cater to diverse requirements and computational resources. In our experiments, we selected the Yolov5s model for its balance of high detection accuracy and reduced computational expense. Furthermore, we set the epoch to 200 and the batch size to 8, with network input images configured at 640 $\times$ 640 pixels.

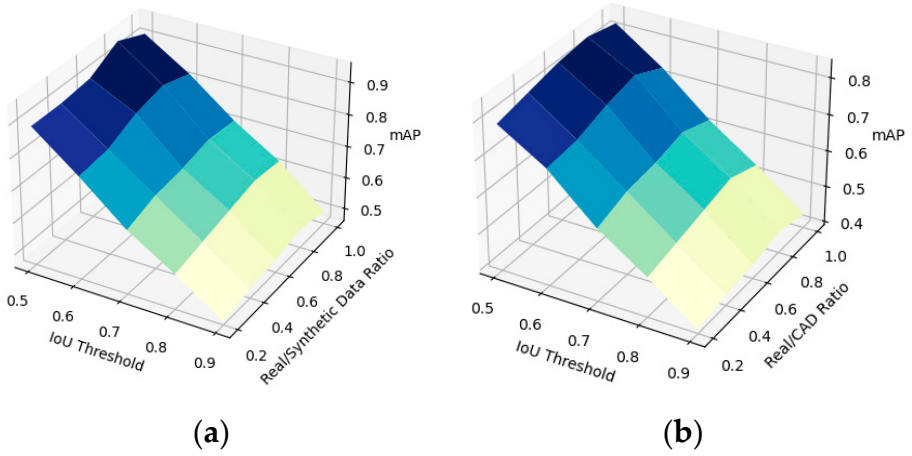
The experimental results demonstrated that the model augmented with the RDB-CycleGAN synthetic images outperformed the model augmented with CAD images, achieving better detection accuracy and generalization ability. This improvement can be attributed to CycleGAN's capability to learn the feature distribution of real images and apply it to synthetic image generation, resulting in synthetic images that closely resemble the distribution of real data. In contrast, CAD images, being computer-generated, might have certain differences from real images, which could lead to inferior performance when used for data augmentation compared to the RDB-CycleGAN synthetic images. The experimental results suggest that CAD images can effectively augment the dataset to some extent, but the synthetic dataset obtained through image translation is more competitive. Therefore, the proposed synthetic dataset is deemed necessary and advantageous. The experimental results are shown in Table 2.

We also controlled the proportions of real data and synthetic data and set different Intersection over Union (IoU) thresholds to obtain the detection accuracy curves under different data ratios and IoU values. The following figures display the curves obtained through varying the proportion of data synthesized using the RDB-CycleGAN network and real data, as well as the curves obtained through varying the proportion of CAD images and real data.

Table 2. The quantitative comparison of the Yolov5 object detection algorithm at a mAP of 0.5 under different amounts of our generated synthetic data, CAD data, and real data preferences, with the best results in bold.

Dataset	Screw	Ubolt	Bolt	mAP@0.5
200 Real Images	0.803	0.783	0.758	0.781
200 Syn Images	0.782	0.811	0.714	0.769
200 Real + 200 CAD	0.823	0.804	0.736	0.787
200 Real + 200 Syn	0.849	0.836	0.795	0.807
200 Real + 400 CAD	0.836	0.817	0.782	0.811
200 Real + 400 Syn	0.877	0.869	0.857	0.867
200 Real + 600 CAD	0.843	0.824	0.806	0.824
200 Real + 600 Syn	**0.910**	**0.878**	**0.862**	**0.883**

In Figure 8, we can observe several trends from the results shown in the above figures. As the IoU changes from 0.9 to 0.5, the mean Average Precision (mAP) gradually increases, with the highest mAP value achieved at an IoU of 0.5. Additionally, when the ratio of real data to synthetic data/CAD images changes, the mAP value also varies. The closer the ratio is to 0.8, the higher the mAP, and the highest mAP value is achieved when the ratio is 0.8. Furthermore, in a cross-comparison, we find that when using our synthetic data, the mAP values are higher than when using CAD images directly under the same conditions. Therefore, we can conclude that our synthetic data demonstrates strong competitiveness, enhancing object detection accuracy and outperforming the use of CAD images.



(**a**) (**b**)

Figure 8. Mean average precision at different image ratios and IoU levels: (**a**) real data and synthetic data; (**b**) real data and CAD image.

The proposed image translation method in this paper was used to synthesize data for three types of industrial parts. We established two sets of training data and conducted tests under the same set of images. The first set of data consisted of 200 real multi-category images and 600 single-category images synthesized using our method (200 images for each of three categories), with the test results shown in Figure 9. The second set comprised 200 real multi-category images and 600 single-category CAD images (200 images per category), with the test results depicted in Figure 10. Both datasets were trained using the Yolov5s model. The detection results indicate a noticeable improvement in accuracy when using our synthesized data.

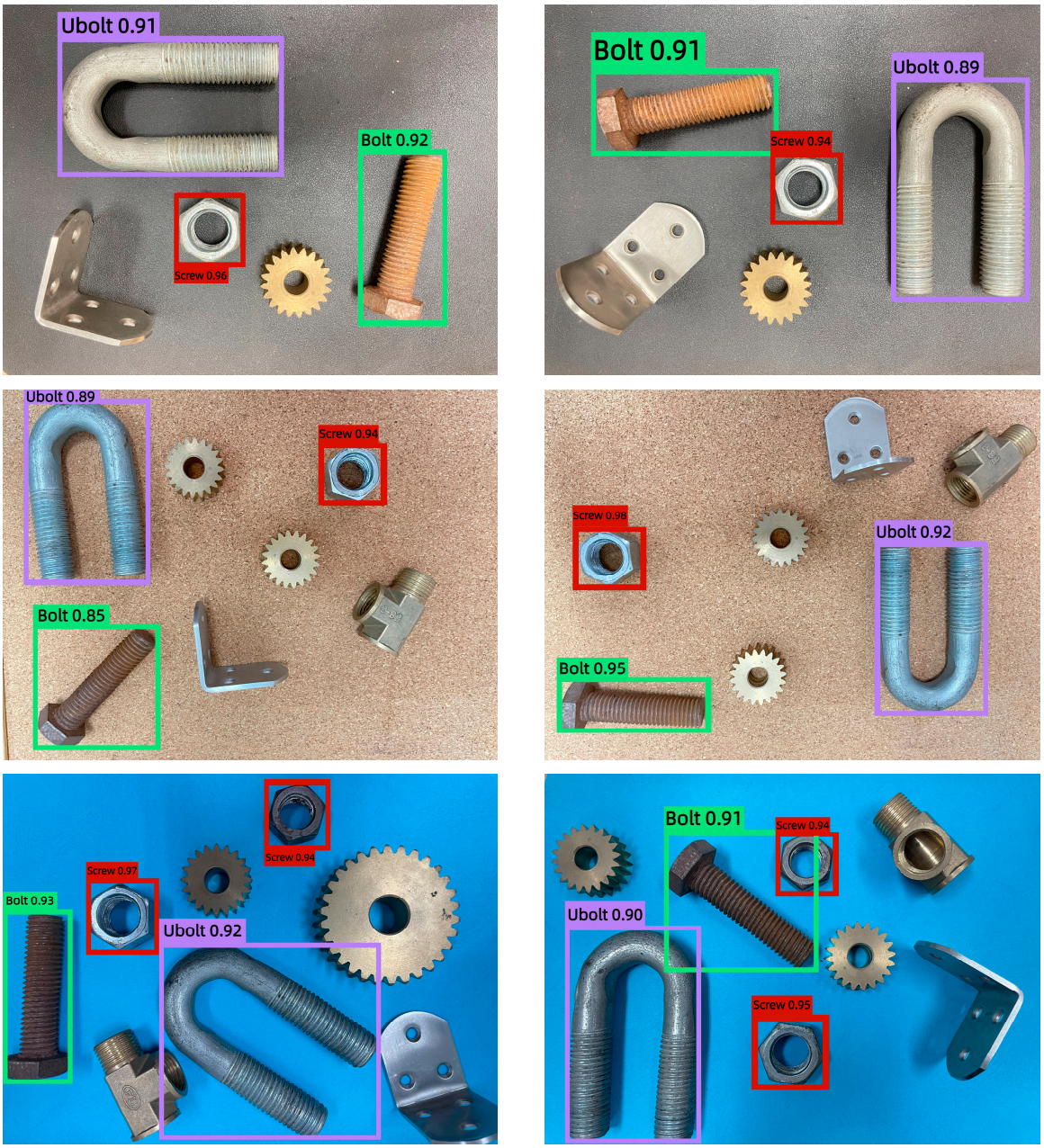


Figure 9. Object detection results trained on real data and our synthetic data.

The results demonstrate that our synthetic data are highly competitive and can effectively augment industrial part data that is difficult to obtain. The following figures display some detection results from the test dataset.

From Figures 9 and 10, we can intuitively observe that within the same group of test images, the detection accuracy using our synthesized data is higher than that achieved with the direct use of CAD images. Through the method of synthesized data proposed in this paper, we can effectively expand the dataset first and foremost. With only a small number of CAD images and real images, we can inexpensively acquire numerous synthesized data. On the other hand, compared to the approach of directly using CAD images to expand the dataset, our synthesized data are more competitive.

Figure 10. Object detection results trained on real data and CAD images.

5. Conclusions

This paper proposes a synthetic data generation framework for object detection tasks, comparing real and synthetic data to analyze how different combinations of real and synthetic data affect the accuracy of object detection models. In the original CycleGAN network, RDB modules and SSIM loss are introduced to improve the quality of synthetic data and complete the translation task from CAD images to real images effectively. In the experimental section, we controlled the ratio of synthetic data to real data, demonstrating that our synthetic data, being directly based on CAD images, effectively augments the dataset and improves detection accuracy.

Limitations and deficiencies: There is still room for further improvement in the quality of our synthetic data. Additionally, our synthetic data are relatively limited in scene diversity, lacking sufficient variation. Future work will focus on further enhancing the

quality of synthetic images and diversifying our synthetic data through incorporating more complex environmental backgrounds.

Author Contributions: Conceptualization, F.X. and J.H.; methodology, F.X. and G.Z.; validation, F.X. and Q.J.; investigation, F.X.; resources, J.H. and P.L.; writing—original draft preparation, F.X.; writing—review and editing, J.H. and Q.J. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Natural Science Foundation of China under Grant No. 52075404 and the Natural Science Foundation of Hubei Province of China under Grant nos. 2023AFB153.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Malburg, L.; Rieder, M.-P.; Seiger, R.; Klein, P.; Bergmann, R. Object detection for smart factory processes by machine learning. *Procedia Comput. Sci.* **2021**, *184*, 581–588. [CrossRef]
2. Zhu, X.; Maki, A.; Hanson, L. Unsupervised domain adaptive object detection for assembly quality inspection. *Procedia CIRP* **2022**, *112*, 477–482. [CrossRef]
3. Liang, B.; Wang, Y.; Chen, Z.; Liu, J.; Lin, J. Object detection and robotic sorting system in complex industrial environment. In Proceedings of the 2017 Chinese Automation Congress (CAC), Jinan, China, 20–22 October 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 7277–7281.
4. Apostolopoulos, I.D.; Tzani, M.A. Industrial object and defect recognition utilizing multilevel feature extraction from industrial scenes with Deep Learning approach. *J. Ambient. Intell. Humaniz. Comput.* **2022**, *14*, 10263–10276. [CrossRef]
5. Kaur, J.; Singh, W. Tools, techniques, datasets and application areas for object detection in an image: A review. *Multimedia Tools Appl.* **2022**, *81*, 38297–38351. [CrossRef] [PubMed]
6. Illarionova, S.; Nesteruk, S.; Shadrin, D.; Ignatiev, V.; Pukalchik, M.; Oseledets, I. Object-based augmentation for building semantic segmentation: Ventura and santa rosa case study. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 11–17 October 2021; pp. 1659–1668.
7. Ghiasi, G.; Cui, Y.; Srinivas, A.; Qian, R.; Lin, T.Y.; Cubuk, E.D.; Le, Q.V.; Zoph, B. Simple copy-paste is a strong data augmentation method for instance segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 2918–2928.
8. Kowalczuk, Z.; Glinko, J. Training of deep learning models using synthetic datasets. In *International Conference on Diagnostics of Processes and Systems*; Springer International Publishing: Cham, Switzerland, 2022; pp. 141–152.
9. Aswar, A.; Manjaramkar, A. Salient Object Detection for Synthetic Dataset. In Proceedings of the International Conference on ISMAC in Computational Vision and Bio-Engineering 2018 (ISMAC-CVB), Palladam, India, 16–17 May 2019; Springer International Publishing: Cham, Switzerland, 2019; pp. 1405–1415.
10. Rajpura, P.S.; Bojinov, H.; Hegde, R.S. Object detection using deep cnns trained on synthetic images. *arXiv* **2017**, arXiv:1706.06782.
11. Bhattacharjee, D.; Kim, S.; Vizier, G.; Salzmann, M. Dunit: Detection-based unsupervised image-to-image translation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 4787–4796.
12. Tang, J.; Zhou, H.; Wang, T.; Jin, Z.; Wang, Y.; Wang, X. Cascaded foreign object detection in manufacturing processes using convolutional neural networks and synthetic data generation methodology. *J. Intell. Manuf.* **2022**, *34*, 2925–2941. [CrossRef]
13. Nowruzi, F.E.; Kapoor, P.; Kolhatkar, D.; Hassanat, F.A.; Laganiere, R.; Rebut, J. How much real data do we actually need: Analyzing object detection per-formance using synthetic and real data. *arXiv* **2019**, arXiv:1907.07061.
14. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial nets. In Proceedings of the Advances in Neural Information Processing Systems, Montreal, QC, Canada, 8–13 December 2014; Volume 27.
15. Jin, Q.; Ma, Y.; Fan, F.; Huang, J.; Mei, X.; Ma, J. Adversarial autoencoder network for hyperspectral unmixing. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *34*, 4555–4569. [CrossRef]
16. Vega-Márquez, B.; Rubio-Escudero, C.; Riquelme, J.C.; Nepomuceno-Chamorro, I. Creation of synthetic data with conditional generative adversarial networks. In Proceedings of the 14th International Conference on Soft Computing Models in Industrial and Environmental Applications (SOCO 2019), Seville, Spain, 13–15 May 2019; Proceedings 14; Springer International Publishing: Cham, Switzerland, 2020; pp. 231–240.

17. Zheng, Z.; Bin, Y.; Lv, X.; Wu, Y.; Yang, Y.; Shen, H.T. Asynchronous generative adversarial network for asymmetric unpaired image-to-image translation. *IEEE Trans. Multimedia* **2022**, *25*, 2474–2487. [CrossRef]
18. Zhang, X.; Fan, C.; Xiao, Z.; Zhao, L.; Chen, H.; Chang, X. Random reconstructed unpaired image-to-image translation. *IEEE Trans. Ind. Inform.* **2022**, *19*, 3144–3154. [CrossRef]
19. Shen, Z.; Huang, M.; Shi, J.; Liu, Z.; Maheshwari, H.; Zheng, Y.; Xue, X.; Savvides, M.; Huang, T.S. CDTD: A large-scale cross-domain benchmark for instance-level image-to-image translation and domain adaptive object detection. *Int. J. Comput. Vis.* **2020**, *129*, 761–780. [CrossRef]
20. Isola, P.; Zhu, J.Y.; Zhou, T.; Efros, A.A. Image-to-image translation with conditional adversarial networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1125–1134.
21. Sultana, M.; Ahmed, T.; Chakraborty, P.; Khatun, M.; Hasan, M.R.; Uddin, M.S. Object detection using template and HOG feature matching. *Int. J. Adv. Comput. Sci. Appl.* **2020**, *11*, 233–238. [CrossRef]
22. Girshick, R. Fast r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
23. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
24. Menke, M.; Wenzel, T.; Schwung, A. Improving gan-based domain adaptation for object detection. In Proceedings of the 2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC), Macau, China, 8–12 October 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 3880–3885.
25. Lin, C.T.; Huang, S.W.; Wu, Y.Y.; Lai, S.H. GAN-based day-to-night image style transfer for nighttime vehicle detection. *IEEE Trans. Intell. Transp. Syst.* **2020**, *22*, 951–963. [CrossRef]
26. Kiefer, B.; Ott, D.; Zell, A. Leveraging synthetic data in object detection on unmanned aerial vehicles. In Proceedings of the 2022 26th International Conference on Pattern Recognition (ICPR), Montreal, QC, Canada, 21–25 August 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 3564–3571.
27. Paulin, G.; Ivasic-Kos, M. Review and analysis of synthetic dataset generation methods and techniques for application in computer vision. *Artif. Intell. Rev.* **2023**, *56*, 9221–9265. [CrossRef]
28. Zhang, H.; Pan, D.; Liu, J.; Jiang, Z. A novel MAS-GAN-based data synthesis method for object surface defect detection. *Neurocomputing* **2022**, *499*, 106–114. [CrossRef]
29. Mishra, S.; Panda, R.; Phoo, C.P.; Chen, C.F.R.; Karlinsky, L.; Saenko, K.; Saligrama, V.; Feris, R.S. Task2sim: Towards effective pre-training and transfer from synthetic data. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 9194–9204.
30. Yang, X.; Fan, X.; Wang, J.; Lee, K. Image translation based synthetic data generation for industrial object detection and pose estimation. *IEEE Robot. Autom. Lett.* **2022**, *7*, 7201–7208. [CrossRef]
31. Arents, J.; Lesser, B.; Bizuns, A.; Kadikis, R.; Buls, E.; Greitans, M. Synthetic Data of Randomly Piled, Similar Objects for Deep Learning-Based Object Detection. In *International Conference on Image Analysis and Processing*; Springer International Publishing: Cham, Switzerland, 2022; pp. 706–717.
32. Rojtberg, P.; Pöllabauer, T.; Kuijper, A. Style-transfer GANs for bridging the domain gap in synthetic pose estimator training. In Proceedings of the 2020 IEEE International Conference on Artificial Intelligence and Virtual Reality (AIVR), Utrecht, The Netherlands, 14–18 December 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 188–195.
33. Liu, W.; Luo, B.; Liu, J. Synthetic data augmentation using multiscale attention CycleGAN for aircraft detection in remote sensing images. *IEEE Geosci. Remote Sens. Lett.* **2021**, *19*, 1–5. [CrossRef]
34. Zhu, J.Y.; Park, T.; Isola, P.; Efros, A.A. Unpaired image-to-image translation using cycle-consistent adversarial networks. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2223–2232.
35. Mohajerani, S.; Asad, R.; Abhishek, K.; Sharma, N.; van Duynhoven, A.; Saeedi, P. Cloudmaskgan: A content-aware unpaired image-to-image translation algorithm for remote sensing imagery. In Proceedings of the 2019 IEEE International Conference on Image Processing (ICIP), Taipei, Taiwan, 22–25 September 2019; IEEE: Piscataway, NJ, USA, 2019.
36. Tang, H.; Bai, S.; Sebe, N. Dual attention gans for semantic image synthesis. In Proceedings of the 28th ACM International Conference on Multimedia, Seattle, WA, USA, 12–16 October 2020.
37. He, J.; Wang, C.; Jiang, D.; Li, Z.; Liu, Y.; Zhang, T. CycleGAN with an improved loss function for cell detection using partly labeled images. *IEEE J. Biomed. Health Inform.* **2020**, *24*, 2473–2480. [CrossRef]
38. He, J.; Wang, C.; Jiang, D.; Li, Z.; Liu, Y.; Zhang, T. Identity-aware CycleGAN for face photo-sketch synthesis and recognition. *Pattern Recognit.* **2020**, *102*, 107249.
39. Huang, S.; Jin, X.; Jiang, Q.; Li, J.; Lee, S.-J.; Wang, P.; Yao, S. A fully-automatic image colorization scheme using improved CycleGAN with skip connections. *Multimed. Tools Appl.* **2021**, *80*, 26465–26492. [CrossRef]
40. Kim, G.; Park, J.; Lee, K.; Lee, J.; Min, J.; Lee, B.; Han, D.K.; Ko, H. Unsupervised real-world super resolution with cycle generative adversarial network and domain discriminator. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, Seattle, WA, USA, 14–19 June 2020; pp. 456–457.
41. Zhang, F.; Gao, H.; Lai, Y. Detail-preserving cyclegan-adain framework for image-to-ink painting translation. *IEEE Access* **2020**, *8*, 132002–132011. [CrossRef]

42. Yi, Z.; Zhang, H.; Tan, P.; Gong, M. Dualgan: Unsupervised dual learning for image-to-image translation. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2849–2857.
43. Yang, S.; Jiang, L.; Liu, Z.; Loy, C.C. Unsupervised image-to-image translation with generative prior. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 18332–18341.
44. Choi, Y.; Uh, Y.; Yoo, J.; Ha, J.W. Stargan v2: Diverse image synthesis for multiple domains. In Proceedings of the IEEE/CVF Conference On Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 8188–8197.