A Semi-supervised Defect Detection Method Based on Image Inpainting

Huibin Cao[1], Yongxuan Lai[1](✉), Quan Chen[2], and Fan Yang[3]

[1] School of Informatics/Shenzhen Research Institute, Xiamen University, Xiamen, China
bboyhb@stu.xmu.edu.cn, laiyx@xmu.edu.cn

[2] Department of Computer Science, Xiamen University Malaysia, Sepang, Malaysia

[3] Department of Automation, Xiamen University, Xiamen, China
yang@xmu.edu.cn

Abstract. Defect detection plays an important role in the industrial field. Because the defective images are often insufficient and defects can be various, defective image synthesis is commonly used and models always tend to learn the distribution of defects. However, the complexity of defective image synthesis and difficulty of detecting unseen defects are still the main challenges. To solve these problems, this paper proposes a semi-supervised defect detection method based on image inpainting, denoted as SDDII, which combines the training strategies of CycleGAN and Pix2Pix. First, we train a defect generator unsupervisedly to generate defective images. Second, we train the defect inpaintor supervisedly using the generated images. Finally, the defect inpaintor is used to inpainting the defects, and the defective areas can be segmented by comparing images before and after inpainting. Without ground truth for training, SDDII achieves better results than the naive CycleGAN, and comparable results with UNET which is supervised learning. In addition, SDDII learns the distribution of contents in defect-free images so it has good adaptability for defects unseen before.

Keywords: Defect detection · Automated optical inspection · Generative adversarial networks

1 Introduction

The defect detection is an important part to ensure product quality. Traditionally, this complex task is completed by manpower which is time-consuming and labor-consuming, where the accuracy is affected by subjective judgments of

This work was supported in part by the Natural Science Foundation of Guangdong under Grant 2021A1515011578; in part by the Natural Science Foundation of China under Grant 61672441 and Grant 61673324; in part by the Natural Science Foundation of Fujian under Grant 2018J01097; in part by the Shenzhen Basic Research Program under Grant JCYJ20170818141325209 and Grant JCYJ20190809161603551.

workers and the efficiency depends on the physical condition of workers. In recent years, with the development of deep learning, more and more defect detection methods based on deep learning have been used to assist or even replace the traditional manpower to improve the accuracy and efficiency [1,3,15].

In industrial scenes, it cost a lot to collect and label defective images so the defective image synthesis is often used to help generate more data. However, defective image synthesis [8] is complicated and not general-purpose which needs to design exclusive method. Additionally, it is still challenging for model to recognize unseen defects even though the data is sufficient.

Recently, to reduce complexity of defective image synthesis, CycleGAN [26] is often used to generate defective images in an image-to-image manner by inputting defect-free images [18]. Through this method, the defective images can be inpainted back to defect-free images conveniently. By comparing the images before and after inpainting, the defective areas can be segmented. This method would make the model to learn the distribution of contents in defect-free images, instead of learning the distribution of defects. It is like to make the model to generate "mind-set", where the unseen contents in images will be inpainted. So this method can be good at detecting unseen defects. However, CycleGAN is trained in an unsupervised manner and the generated defective images are not utilized. It still has improvement space.

The main contribution of this paper is that we propose a semi-supervised method utilizing the generated defective images to further improve inpainting performance of CycleGAN [26]. Our method is denoted as SDDII, which stands for "Semi-supervised Defect Detection based on Image Inpainting". Firstly, CycleGAN is trained to generate the defective images and inpaint the defective images. Secondly, we introduce the training strategy of Pix2pix [7], which utilizes the generated defective images to supervisedly train the defect inpainting. Finally, the defects are segmented by comparing the images before and after inpainting. Experiments show that SDDII can achieve better results than the naive CycleGAN, and comparable results with the UNET which is supervised learning.

The rest of the paper is structured as follows: in Sect. 2, we review the literature examining the applications of GAN on defect detection. In Sect. 3 we outline the methodology employed, while in Sect. 4 we report the experiments and results. In Sect. 5, we make a brief summary of this paper.

2 Related Work

GAN [6] is a network proposed by Ian Goodfellow in 2014. Compared with other applications of convolutional network, such as image classification [19], object detection [4], semantic segmentation [16], GAN can generate new data by inputting random noise. Applications of GAN include face generation [23], super-resolution generation [25], image inpainting [10], etc. To some extent, GAN is a way of using reinforcement learning [9] to realize generative tasks. The ideas of GAN and the "Actor-Critic" algorithm in reinforcement learning [12] are similar, where actors are like generators in GAN, and critics are like discriminators in GAN, maintaining a game between actors (generators) and critics (discriminators) to achieve Nash equilibrium [20].

As GAN becomes more and more popular, GAN is gradually introduced into the field of defect detection [2,14,24]. The common application is defective image synthesis where defects are generated through GAN and pasted into the defect-free image. Defective image synthesis has the following disadvantages: the design process is complicated and the pasted images look fake. To overcome these, we use CycleGAN [26] to generate the defective images in an image-to-image manner where a forward mapping flow and a backward mapping flow form a circle. In this method [18], a generator in a CycleGAN is adopted to generate defective images first, then the other generator is used to inpaint defective images, finally the defective areas can be segmented by comparing the images before and after inpainting (as shown in Fig. 1). However, consistency loss ($LossF2$) and the adversarial loss ($LossB1$) are in two separated mapping flows which leads to an imbalance in the training of DI (Defect Inpaintor). The performance of DI has a direct impact on defect detection, so we aim to further improve the defect inpainting. At this research, we introduce the supervised training strategy of Pix2pix [7], where the consistency loss ($LossF4$) and the adversarial loss ($LossF3$) are combined together in forward mapping flow (as shown in Fig. 2) to improve the defect inpainting.

3 Methodology

3.1 Architecture

This paper proposes a semi-supervised defect detection method, denoted as SDDII, which combines the training strategies of CycleGAN [26] and Pix2Pix [7]. The training strategy of SDDII can be divided into two phases: the first phase is unsupervised phase which contains unsupervised defect generation and unsupervised defect inpainting. The second phase is supervised phase in which we train additional supervised defect inpainting.

The unsupervised phase is implemented with CycleGAN, as shown in Fig. 1. Through CycleGAN, data do not need to be collected in pairs [26]. Two generators in CycleGAN are defined as DG (Defect Generator) and DI (Defect Inpaintor), which are responsible for defect generation and defect inpainting respectively. We define the defect-free dataset as X and the defective dataset as Y. As shown in Fig. 1, a defect-free image x in X is used to generate a defective image $DG(x)$ by DG, then $DG(x)$ is inpainted back to a defect-free image $DI(DG(x))$ by DI. In the same way, a defective image y in Y is inpainted to a defect-free image $DI(y)$ by DI, then the $DI(y)$ is used to generate a defective image $DG(DI(y))$ by DG.

It is worth noting that in the actual operation process, the performance of defect generation is often better than that of defect inpainting. That is because the generated defects do not need too many constraints generally. The generated content can be considered as defects as long as it can "destroy" the defect-free images to some extent. As for the defect inpainting, we not only need to inpaint

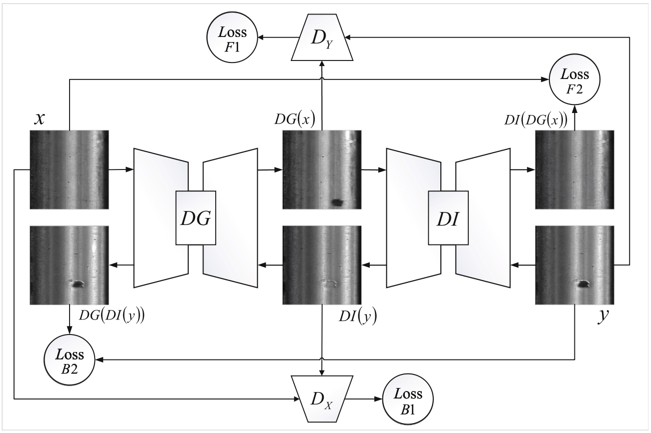

Fig. 1. The unsupervised training phase of SDDII. We train CycleGAN to generate defective images and inpaint defects. Two generators in CycleGAN are defined as DG (Defect Generator) and DI (Defect Inpaintor), which are responsible for defect generation and defect inpainting respectively. The loss functions in the forward mapping flow are defined as $LossF$x, while those in the backward mapping flow are defined as $LossB$x.

the defective content in local area, but also need to make the content distribution of the whole image close to that of the defect-free images. So in practice, the difficulty of defect inpainting is greater than that of defect generation.

Since CycleGAN [26] can use unpaired data to generate paired data, we introduce an additional supervised training phase to improve the defect inpainting. As shown in Fig. 2, we fix the parameters of DG and then input a defect-free image x to generate a defective images $DG(x)$. x and $DG(x)$ are used for supervised training of DI. In addition, a new discriminator D_Z is added to form adversarial training. The S operation is channel stacking, where the defective images $DG(x)$ are stacked with the defect-free images x or inpainted images $DI(DG(x))$, the stacked images will be inputted to the discriminator for comparison. In essence, it is an implementation of conditional GAN [17], which is also used in the famous Pix2pix [7] and has achieved good results.

In terms of model structure, we adapt both DG and DI from UNET [21]. D_X, D_Y and D_Z use modules of the discriminator structure of patchGAN [7] while the number of input channels of D_Z is twice that of D_X and D_Y. For example, for RGB color images, the number of input channels of D_X and D_Y is 3, and that of D_Z is 6 because the S operation stacks two images with channel number of 3 into a feature map with channel number of 6.

In the test phase, as shown in Fig. 3. The defective images y are inpainted to defect-free images $DI(y)$ by DI. Then the images before and after the inpainting, y and $DI(y)$, are compared pixel by pixel and the pixels whose absolute

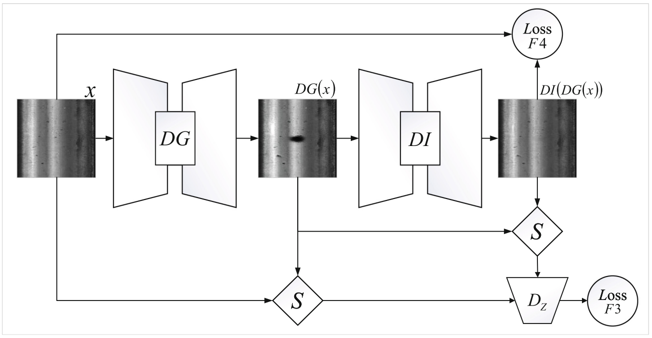

Fig. 2. The supervised training phase of SDDII. We introduce an additional supervised training phase which aims to improve defect inpainting. The parameters of DG are fixed to generate defective images for DI's training. S denotes image channel stacking.

difference is greater than the threshold will be segmented to form a binary image $D\&T(y, DI(y))$. The binary image $D\&T(y, DI(y))$ represents the locations and shapes of the defects, where $D\&T$ denotes pixel-level difference computing and thresholding.

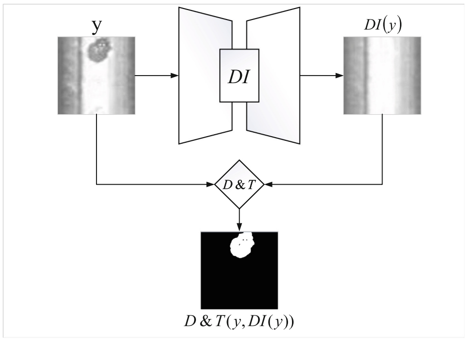

Fig. 3. The test phase of SDDII. DI is used to inpaint the images in testset. And then we compare the images before and after inpainting pixel by pixel to get the defective area, that is to calculate the prediction mask $D\&T(y, DI(y))$. $D\&T$ denotes pixel-level difference computing and thresholding.

3.2 Loss Function

Unsupervised Phase. In the unsupervised phase, which contains defect generation and inpainting, the loss is divided into two parts, the adversarial loss and the consistency loss. As shown in Fig. 1, there are the forward $(X \rightarrow Y \rightarrow X)$ mapping flow and the backward $(Y \rightarrow X \rightarrow Y)$ mapping flow. Among them, $X \rightarrow Y$ in the forward mapping flow is completed by DG cooperating with the discriminator D_Y, and the loss function is shown in the Equation LossF1. DG

tries to minimize this function against D_Y who tries to maximize it, that is, $\min_{DG}\max_{D_Y} L_{F1}(DG, D_Y)$. Similarly, for $Y \rightarrow X$ in the backward mapping flow, the loss function is shown in the Eq. 2, i.e., $\min_{DI}\max_{D_X} L_{B1}(DI, D_X)$. The loss functions in the forward mapping flow are defined as $LossF$x, and those in the backward mapping flow are defined as $LossB$x. We define the data distribution as $x \sim Pdata(x)$ and $y \sim Pdata(y)$ where $Pdata$ denotes the empirical distribution of dataset.

$$\begin{aligned} L_{F1}(DG, D_Y) = & E_{y\sim Pdata(y)}[\log D_Y(y)] \\ & + E_{x\sim Pdata(x)}[\log(1 - D_Y(DG(x)))] \end{aligned} \tag{1}$$

$$\begin{aligned} L_{B1}(DI, D_X) = & E_{x\sim Pdata(x)}[\log D_X(x)] \\ & + E_{y\sim Pdata(y)}[\log(1 - D_X(DI(y)))] \end{aligned} \tag{2}$$

$Y \rightarrow X$ in the forward mapping flow is mapped by DI. And $X \rightarrow Y$ in the backward mapping flow is mapped by DG. The consistency loss of each of them corresponds to $LossF2$ and $LossB2$ in Fig. 1 respectively. The loss functions are shown in the Eq. 3 and the Eq. 4.

$$L_{F2}(DG, DI) = E_{x\sim Pdata(x)}[\|DI(DG(x)) - x\|_1] \tag{3}$$

$$L_{B2}(DG, DI) = E_{y\sim Pdata(y)}[\|DG(DI(y)) - y\|_1] \tag{4}$$

The total loss function of the unsupervised phase is shown in the Eq. 5, where α is a balance parameter between adversarial loss and consistency loss. In this phase, we aim to solve $DG^*, DI^* = \arg\min_{DG,DI}\max_{D_X,D_Y} L_{phase1}(DG, DI, D_X, D_Y)$.

$$\begin{aligned} L_{phase1}(DG, DI, D_X, D_Y) = & L_{F1}(DG, D_Y) + L_{B1}(DI, D_X) \\ & + \alpha(L_{F2}(DG, DI) + L_{B2}(DG, DI)) \end{aligned} \tag{5}$$

Supervised Phase. In the supervised phase, the parameters of DG are fixed and only the DI are trained because we aim to improve the defect inpainting. As shown in Fig. 2, the defect-free image x is inputted to DG to generate the defective image $DG(x)$, which is then inpainted by DI to obtain $DI(DG(x))$. The discriminator D_Z compares and judges the images before and after inpainting, so as to assist DI to optimize the inpainting performance. In the training phase, the discriminator label is true for the stacking input of x and $DG(x)$, and false for the stacking input of $DI(DG(x))$ and $DG(x)$. And the loss is also divided into adversarial loss and consistency loss. The adversarial loss corresponds to $LossF3$ in Fig. 2, as shown in the Eq. 6 where z and Pz denote Gaussian noise and Gaussian distribution respectively. According to cGAN [17], we should provide Gaussian noise z as an input to the generator in addition to original input. If z is ignored, the mapping still can be learned, however, deterministic outputs will be produced and DI can not learn a whole distribution.

$$\begin{aligned} L_{F3}(DI, D_Z) = & E_{x\sim Pdata(x)}[\log D_Z(DG(x), x)] \\ & + E_{x\sim Pdata(x), z\sim Pz(z)}[\log(1 - D_Z(DG(x), DI(DG(x), z)))] \end{aligned} \tag{6}$$

The consistency loss can be calculated according to the pixel difference between the defect-free image x and the inpainted image $DI(DG(x))$, corresponding to $LossF4$ in Fig. 2, as shown in the Eq. 7.

$$L_{F4}(DI) = E_{x\sim Pdata(x),z\sim Pz(z)}[\|DI(DG(x),z) - x\|_1] \tag{7}$$

The total loss function of the supervised phase is shown in the Eq. 8, in which β is a balance parameter between the adversarial loss and the consistency loss. In this phase, we aim to solve $DI^* = \arg\min_{DI}\max_{D_Z} L_{phase2}(DI, D_Z)$.

$$L_{phase2}(DI, D_Z) = L_{F3}(DI, D_Z) + \beta L_{F4}(DI) \tag{8}$$

4 Experiments

4.1 Preparations

We evaluate SDDII on the RSDDs (Rail Surface Defect Datasets) [5] which contains two types of datasets. The first is a Type-I RSDDs captured from express rails, which contains 67 challenging images. The second is a Type-II RSDDs captured from common/heavy haul rails, which contains 128 challenging images. Each image contains at least one defect, and there is a corresponding ground truth, which is a binary mask image with the same size that shows the locations and shapes of the defects.

For evaluation, we choose IOU (Intersection over Union), pixel-level precision, pixel-level recall, and pixel-level F1 score as the evaluation metrics which are commonly used in semantic segmentation. As shown in Eq. 9, TP, FP, and FN denote the numbers of correctly detected pixels, falsely detected pixels, and undetected defect pixels. The values in following experiments are averaged over the whole testset.

$$\begin{aligned} IOU &= TP/(TP + FP + FN) \\ Pre &= TP/(TP + FP) \\ Rec &= TP/(TP + FN) \\ F1 &= 2 \times Pre \times Rec/(Pre + Rec) \end{aligned} \tag{9}$$

For comparison, on these datasets we also implement naive CycleGAN [26] for ablation study, and UNET [21] which needs binary mask images for supervised training while SDDII does not need. To be fair, no pre-trained model is used in our experiments.

4.2 Implementation Details

For Type-I RSDDs, the width of each image is 160 px, and the height can be from 1000 to 1282 px. And for Type-II RSDDs, all images share the same size of 55×1250 px^2. To save GPU memory, we crop the original images and mask images via sliding windows. What's more, by this trick we can get more training images although the number of images in original dataset is small. In our experiments, the window size is set to 160×160 px^2 and 55×55 px^2, and the sliding stride is set to 80 and 27 px, for Type-I and Type-II RSDDs respectively. All the cropped images will be resized to 256×256 px^2 before being inputted. As for labelling, if a cropped binary image contains any TRUE pixels(pixel value equals 1), we label the corresponding cropped image as defective image, otherwise, defect-free image.

During the unsupervised training phase, we input the cropped images to train our model with Adam [11] solver from scratch. The α is set to 10 and the batchsize is set to 1. In the first 100 epoch we keep the learning rate at 0.0002 and in the next 100 epoch we decay the learning rate to zero linearly.

During the supervised training phase, we randomly pick a image x from X and input it to DG to generate defective image $DG(x)$, where x and $DG(x)$ make up the paired images for supervised training of DI. In theory, we can generate training paired images infinitely. In our experiments, we generate 10000 paired images for DI's further training and we set $\beta = 100$ then keep other training options the same as those of the unsupervised training phase.

In the test phase, we crop each image into an image-set and input the cropped images into DI, like what we do in the training phase. For each cropped image and its output, we compute the difference of each pixel between them and get a pixel-level difference image. After inference of each image-set, we gather the difference images and rebuild the whole prediction binary image which is used to be compared with ground truth for evaluation.

4.3 Results

We train our model on a machine with a GPU of NVIDIA GTX 1660 Ti @ 6GB and a CPU of Intel(R) Core(TM) i7-3770 CPU @ 3.40 GHz. It takes 2–3 days to complete all the trainings. Then we implement the inference and the results are shown in the Table 1 and Table 2. In addition, we also refer to the results of CFE [5] which is proposed by the RSDDs publisher. By now, CFE still achieves the best results. However, CFE is an exclusive system which is specially designed for RSDDs, which contains lots of designs of feature extracting. That is, if we change the dataset, we should design the method again, which is complicated and not general-purpose.

Due to the complexity of analyzing the performance of inpainting, we directly analyze the performance of segmentation instead. Besides, the performance of segmentation is representative of the performance of inpainting.

As for UNET, we can see its performance differs a lot between two datasets. On Type-I RSDDs the most of defects tend to be oval-shaped but on Type-II RSDDs the shapes of defects are various. UNET is good at learning invariant patterns so it get better results than SDDII on Type-I RSDDs, but get worse

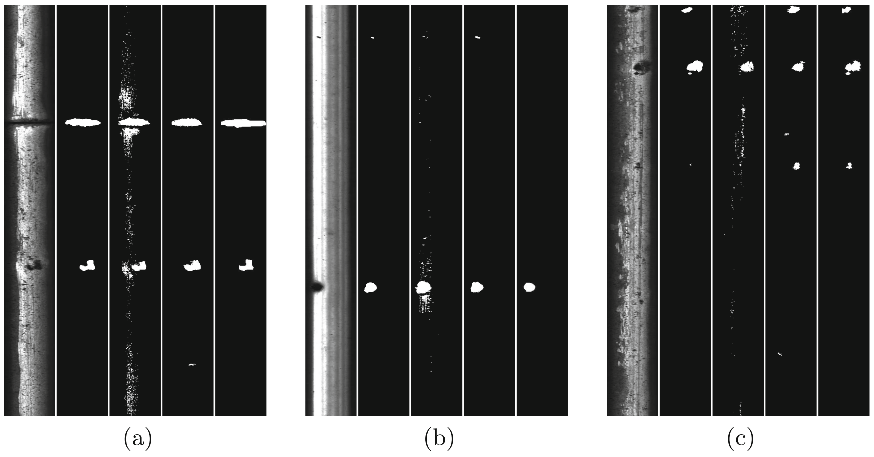

Fig. 4. The results of Rail 40 (a), 54 (b), and 55 (c) in Type-I RSDDs. The sequence of images for each rail is the original image, result of UNET, result of CycleGAN, result of SDDII, ground truth.

IOU, *Pre* and *F*1 than SDDII on Type-II RSDDs. Relatively, SDDII get more stable performances on both Type-I and Type-II RSDDs. SDDII detects defects by comparing images before and after inpainting, so it is adaptive for those unseen defects and can handle defects with various shapes in Type-II RSDDs. Besides, UNET is supervised learning which needs ground truth for training while SDDII does not need. Some representative results are picked and shown in Fig. 4 and Fig. 5. We can see that SDDII output comparable mask with UNET on Type-I RSDDs in Fig. 4 and get more stable and accurate outputs of Type-II RSDDs in Fig. 5.

Table 1. Results on Type-I RSDDs

Method	IOU(%)	Pre(%)	Rec(%)	F1(%)
CFE [5]	–	87.54	85.63	85.12
UNET [21]	45.73	89.25	58.33	68.86
CycleGAN [26]	27.18	60.25	46.40	48.45
SDDII (Ours)	**36.50**	**83.42**	**48.15**	**57.04**

Compared to CycleGAN which is unsupervised learning, SDDII is semi-supervised learning where we additionally implement the supervised training by utilizing the generated paired data. And as shown in Table 1 and Table 2, SDDII outperforms CycleGAN on both Type-I RSDDs and Type-II RSDDs, which shows that our proposed supervised training phase can significantly improve

Table 2. Results on Type-II RSDDs

Method	IOU(%)	Pre(%)	Rec(%)	F1(%)
CFE [5]	–	83.88	83.58	82.11
UNET [21]	24.57	63.49	52.72	50.91
CycleGAN [26]	21.11	64.48	37.57	42.57
SDDII (Ours)	**26.82**	**70.72**	**49.75**	**54.48**

the defect inpainting and then improve the defect detection of CycleGAN. As shown in Fig. 4 and Fig. 5, SDDII's outputs are more stable and accurate than that of CycleGAN.

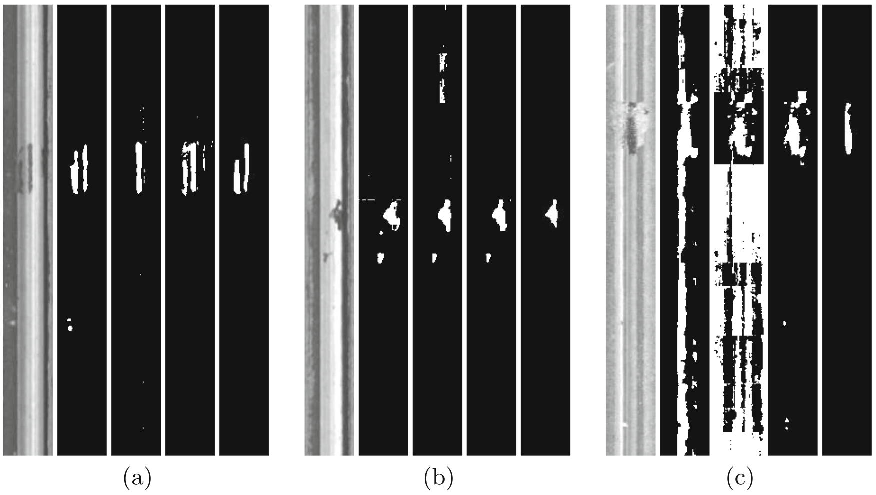

Fig. 5. The results of Rail 6 (a), 11 (b), and 40 (c) in Type-II RSDDs. The sequence of images is the same as Fig. 4.

5 Conclusions

Aiming at reducing the complexity of defective image synthesis and difficulty of detecting unseen defects, we proposed a semi-supervised method, denoted as SDDII, which combines the training strategy of CycleGAN and Pix2Pix. First, through unsupervised training phase, DG and DI would get the abilities of defect generation and defect inpainting respectively. Second, through supervised training phase, we further improve the defect inpainting of DI. Finally, experiments show that SDDII can indeed achieve better results than naive CycleGAN. In addition, SDDII is practical in many industrial scenes, since SDDII has good adaptability for unseen defects and we do not need to label the segmentation masks for training.

References

1. Cao, W., Liu, Q., He, Z.: Review of pavement defect detection methods. IEEE Access **8**, 14531–14544 (2020)
2. Cui, Z., Zhang, M., Cao, Z., Cao, C.: Image data augmentation for sar sensor via generative adversarial nets. IEEE Access **7**, 42255–42268 (2019)
3. Czimmermann, T., Ciuti, G., Milazzo, M., Chiurazzi, M., Roccella, S., Oddo, C.M., Dario, P.: Visual-based defect detection and classification approaches for industrial applications–a survey. Sensors **20**(5), 1459 (2020)
4. Dhillon, A., Verma, G.K.: Convolutional neural network: a review of models, methodologies and applications to object detection. Prog. Artif. Intell. **9**(2), 85–112 (2019). https://doi.org/10.1007/s13748-019-00203-0
5. Gan, J., Li, Q., Wang, J., Yu, H.: A hierarchical extractor-based visual rail surface inspection system. IEEE Sens. J. **17**(23), 7935–7944 (2017)
6. Goodfellow, I.J., et al.: Generative adversarial networks. arXiv:1406.2661 (2014)
7. 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, pp. 1125–1134 (2017)
8. Jain, S., Seth, G., Paruthi, A., Soni, U., Kumar, G.: Synthetic data augmentation for surface defect detection and classification using deep learning. J. Intell. Manuf. 1–14 (2020). https://doi.org/10.1007/s10845-020-01710-x
9. Kaelbling, L.P., Littman, M.L., Moore, A.W.: Reinforcement learning: a survey. J. Artif. Intelli. Res. **4**, 237–285 (1996)
10. Kim, J., Tae, D., Seok, J.: A survey of missing data imputation using generative adversarial networks. In: 2020 International Conference on Artificial Intelligence in Information and Communication (ICAIIC), pp. 454–456. IEEE (2020)
11. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. arXiv:1412.6980 (2014)
12. Konda, V.R., Tsitsiklis, J.N.: Actor-critic algorithms. In: Advances in Neural Information Processing Systems, pp. 1008–1014. Citeseer (2000)
13. Li, X., Su, H., Liu, G.: Insulator defect recognition based on global detection and local segmentation. IEEE Access **8**, 59934–59946 (2020)
14. Lian, J., Jia, W., Zareapoor, M., Zheng, Y., Luo, R., Jain, D.K., Kumar, N.: Deep-learning-based small surface defect detection via an exaggerated local variation-based generative adversarial network. IEEE Trans. Industr. Inf. **16**(2), 1343–1351 (2019)
15. Luo, Q., Fang, X., Liu, L., Yang, C., Sun, Y.: Automated visual defect detection for flat steel surface: a survey. IEEE Trans. Instrum. Meas. **69**(3), 626–644 (2020)
16. Minaee, S., Boykov, Y.Y., Porikli, F., Plaza, A.J., Kehtarnavaz, N., Terzopoulos, D.: Image segmentation using deep learning: a survey. IEEE Transactions on Pattern Analysis and Machine Intelligence (2021)
17. Mirza, M., Osindero, S.: Conditional generative adversarial nets. arXiv:1411.1784 (2014)
18. Niu, S., Li, B., Wang, X., Lin, H.: Defect image sample generation with gan for improving defect recognition. IEEE Trans. Autom. Sci. Eng. **17**(3), 1611–1622 (2020)
19. Obaid, K.B., Zeebaree, S., Ahmed, O.M., et al.: Deep learning models based on image classification: a review. Int. J. Sci. Bus. **4**(11), 75–81 (2020)
20. Pfau, D., Vinyals, O.: Connecting generative adversarial networks and actor-critic methods. arXiv:1610.01945 (2016)

21. Ronneberger, O., Fischer, P., Brox, T.: U-net: convolutional networks for biomedical image segmentation. In: International Conference on Medical Image Computing And Computer-assisted Intervention, pp. 234–241. Springer (2015). https://doi.org/10.1007/978-3-319-24574-4_28
22. Suh, S., Lukowicz, P., Lee, Y.O.: Fusion of global-local features for image quality inspection of shipping label. arXiv:2008.11440 (2020)
23. Tolosana, R., Vera-Rodriguez, R., Fierrez, J., Morales, A., Ortega-Garcia, J.: Deepfakes and beyond: a survey of face manipulation and fake detection. Inf. Fus. **64**, 131–148 (2020)
24. Wang, G., Kang, W., Wu, Q., Wang, Z., Gao, J.: Generative adversarial network (gan) based data augmentation for palmprint recognition. In: 2018 Digital Image Computing: Techniques and Applications (DICTA), pp. 1–7. IEEE (2018)
25. Wang, Z., Chen, J., Hoi, S.C.: Deep learning for image super-resolution: a survey. IEEE transactions on pattern analysis and machine intelligence (2020)
26. 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, pp. 2223–2232 (2017)