Assigned MURA Defect Generation Based on Diffusion Model

Weizhi LIU
TCL Corporate Research (Hong Kong)
wzliu@tcl.com

Chang LIU
The University of Hong Kong
u3603472@connect.hku.hk

QIANG LIU
TCL Corporate Research (Hong Kong)
qiang51.liu@tcl.com

Dahai Yu
TCL Corporate Research (Hong Kong)
dahai.yu@tcl.com

Abstract

In this paper, we propose a novel method for assigned MURA generation using a diffusion model. MURA is a well-known problem in the display industry, which is difficult to be inspected because it is characterized by low contrast, blurry contours, blocky uneven brightness, and irregular shape patterns, and most defects have no rules to follow. Especially, for data-driven deep learning, the shortage of MURA samples collecting from the pipeline of manufactory is the first challenging problem, because the MURA sample happens with a low probability and in various ways. To relieve the problem, our proposed approach employs a diffusion model that generates MURA defect images using a few samples, which allows us to assign the position and class of MURA in the image. Specifically, our method leverages the diffusion process to estimate the visibility of MURA, which is then used to enhance the flexibility of the MURA detection process. We evaluate the performance of our method through MURA inspection. The results demonstrate the effectiveness of our proposed approach in addressing the MURA detection problem.

1. Introduction

Mura is a visual defects phenomenon happened on the Liquid crystal displays (LCD) such as uneven brightness, color variations, or visible defects like spots, lines, or blobs, which may cause unpleasant feelings in applications where visual quality is critical such as in high-end televisions, smartphones, and other electronic devices. Some of the spot examples are shown in Figure1. In the display industry, Mura inspection is an important step in ensuring the quality of display panels. The detection and classification of mura defects can help prevent defective displays from reaching consumers and can also help manufacturers identify and address production issues. However. it is a very challenging task for MURA inspection [28], which is attributed to the properties of MURA: (1) low contrast and uneven brightness [10]; (2) no fixed size and regular shape; (3) resulted from multiple pipelines and diversity products. Additionally, the shortage and imbalance of defect data from factories increase the difficulty of the task. Fortunately, with the progress of computer vision technology, Mura inspection based on deep learning is becoming increasingly promising in industry research.

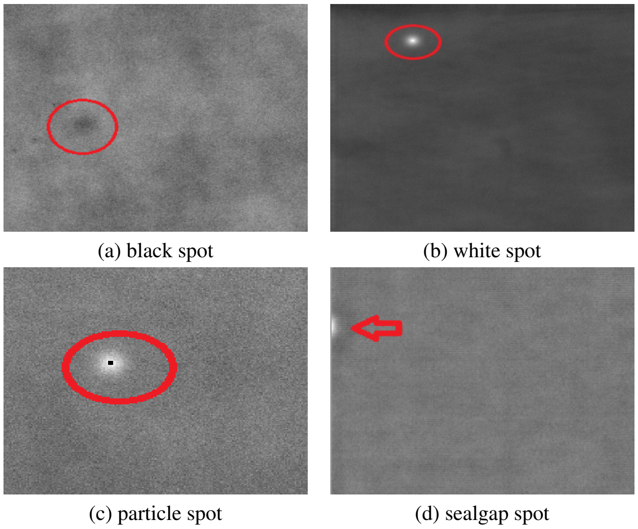

(a) black spot
(b) white spot
(c) particle spot
(d) sealgap spot

Figure 1. Some examples of spot MURAs are marked by red circles or arrows. White spot and black spot are fulfilled circle regions in different color patterns. The white spot is fulfilled with white, while the black spot is fulfilled with black. Particle spots have a bright/black dot in the center of the black/white mura. Sealgap is a type of MURA located on the border of the product. So it is a half-spot.

Mura inspection is typically performed using specialized equipment such as automated optical inspection systems, which use cameras and image processing technology to detect and analyze defects. Image processing-based MURA inspection generally can be divided into two categories [28]: pixel-wise and feature-level. The former can represent the complete information of the LCD panel image, and the recognition accuracy is high but the speed is slow resulting from massive data. The latter are just some features of the image data, such as contrast, area, position, edge, shape, and gray uniformity. These features cannot completely represent the image information, and the recognition accuracy will decrease. Recent work are focusing on a). extracting distinguishable information features from the original defect image, such as YoloV5 [19]; and b). generate defect data, such as Generative Networks [11,14,32,41].

In order to build a robust feature model, Collecting and annotating a large amount of labeled data is inevitable for deep learning especially. This is because deep learning models are designed to learn from data by extracting patterns and relationships by optimization. To optimize effectively, it needs to be trained on a diverse set of examples that cover the range of situations. But collecting data from a factory and annotating data professionally is challenging for several reasons. Firstly, the data is spread across different systems, machines, or devices that are not interconnected, making it difficult to gather all the necessary data in one place. Secondly, the data may not be in a usable format for machine learning algorithms. Data may need to be cleaned, transformed, or normalized before it can be used to train a model. Thirdly, It has a very low probability of a MURA defect occurring in the manufacturing industry as it depends on various factors such as the type of material being manufactured, the manufacturing process, and the quality control measures in place. Fourthly, there may be privacy and security concerns around collecting and using data from a factory. It is important to ensure that any data collected is done in a way that is compliant with relevant regulations and does not compromise the safety or privacy of the company. Lastly and most importantly, the data should be annotated by professional inspectors who have been well-trained and experienced in production quality assessment from different domains.

To overcome these challenges, it is essential to have a well-defined data generation strategy that takes into account the specific requirements of the factory and the machine learning algorithm being used. This strategy for data generation should be used by existing machine learning algorithms, consistent with real data, compatible crossing-platform, automatically annotated data, and no privacy or security concerns. Based on the above requirements, we propose Assigned MURA Defect Generation Based on Diffusion Model, which can help to improve the performance of a deep learning model for detection, but it is important to ensure that the generated data is representative of the real data and does not introduce biases or artifacts that could negatively impact the model's performance.

Comparing the similarity and differences between defect generation and inpainting tasks, we propose novel methods for mura generation with controllable features including but not limited to position, size, and visibility. Our contribution includes 1) the exploration of defect generation using diffsion model, 2) defect dataset expansion through conditions, and 3) texture reserving for better background compatibility.

The rest of the paper is organized as follows. Section 2 provides an overview of the related work in MURA generation and diffusion models. Section 3 describes the proposed approach in detail, including the diffusion model based image generation and feature Reservation. Section 4 presents the experimental results and performance evaluation. Finally, Section 5 concludes the paper and discusses future directions for research.

2. Related Work

Recently, for many image-to-image tasks [17], Generative Adversarial Networks (GANs) [11,32], the most popular research topics in the past few years, are capable of generating high fidelity outputs, broadly applicable, and support efficient sampling. Autoregressive Models [31, 46], VAEs [23, 44], and Normalizing Flows [9, 21] have seen success in specific applications. Nevertheless, GANs can be challenging to train [1, 12], and often suffer from mode collapse [27,33].

In contrast, diffusion and score-based models [14,39,41] have a stable training process and provide more diversity [2, 4, 16, 22, 42, 45], resulting in several key advances in modeling data from different domain. Diffusion models [5, 6, 24] on audio synthesis have achieved human evaluation scores on par with SoTA autoregressive models. On the class-conditional ImageNet generation challenge, [8,15] have outperformed strong GAN baselines in terms of FID scores. Image super-resolution [35], unpaired image-to-image translation [37], and image editing [26,38] also have extensively researched on diffusion model.

Denoising Diffusion Probabilistic Models (DDPM) [14], as a special kind of variational auto-encoders (VAEs) [23], provides a high quality image synthesis method through an iterative denoising process, where the forward diffusion stage corresponds to the encoding process inside VAE [7], while the reverse diffusion stage corresponds to the decoding process. [34] proposed a framework for image-to-image translation using diffusion models, which focus on multiple tasks, colorization, inpainting, uncropping, and JPEG restoration [7]. The aim of [47] is to improve current image-to-image translation score-based diffusion models by utilizing data from a source domain with an equal significance, which leads to generating images that preserve the domain agnostic features while translating characteristics specific to the source domain to the target domain.

Despite these advantages, diffusion models are still inefficient when compared to GANs, requiring multiple network evaluations during inference. Song et al. [40] proposed Denoising Diffusion Implicit Models (DDIM), which formulates an alternative non-Markovian noising process that has the same forward marginals as DDPM, but allows producing different reverse samplers by changing the variance of the reverse noise. By setting this noise to 0, they provide a way to turn any model into a deterministic mapping from latents to images, and find that this provides an alternative way to sample with fewer steps. We adopt this sampling approach when using fewer than 100 sampling steps, since Nichol and Dhariwal [30] found it to be beneficial in this regime. Our proposed diffusion models build on these recent advances, showing versatility on a suite of image-to-image translation tasks.

3. Methodologies

Our work is based on [34]. The model we are using is U-Net architecture from guided diffusion, and the attention mechanism is applied in low-resolution features as valilla DDPM. In this section, we illustrate step by step how we apply a diffusion process to MURA generation.

3.1. Image Generation with DDPM

Recall that in DDPM process, the forward process is denoted as:

$$q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{\beta_t}x_{t-1}, (1-\beta_t)\epsilon) \tag{1}$$

which is parametererized by $\beta = \beta_1 ... \beta_t$ series that evolve over time according to a fixed or learned schedule and $\epsilon \sim \mathcal{N}(0,1)$. And x_t, originating from x_0, the original input, represents the sampled value at time step t. Suppose for each time step, we set α_t that equals $1 - \beta_t$, and

$$\bar{\alpha_t} = \prod_{i=1}^{t} \alpha_i \tag{2}$$

then because of the Gaussian process, for any time step t, x_t can be calculated in a closed form:

$$x_t = \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon \tag{3}$$

For the backward pass, we can try to minimize the loss for either means, noisy values, or the score functions between the sampled value $x_t \sim q(x_t|x_{t-1})$ and the predicted value $\hat{x}_t \sim p(x_t|x_{t+1})$. Taking predicting the noise for example, for each backward step, we use the model to predict ϵ from x_t, followed by the calculation of x_0 and x_{t-1} respectively:

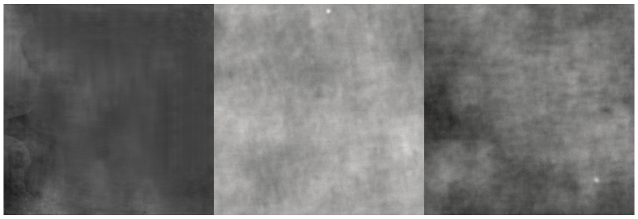


Figure 2. Images with mura defects generated using DDPM.

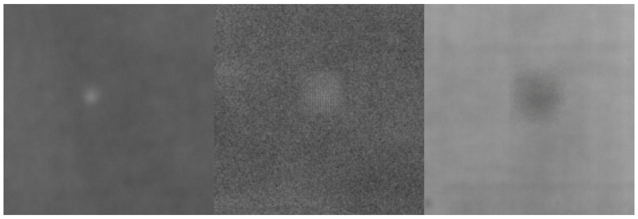


Figure 3. Conditional image, masked with random value, which is the original implementation. The defect region is extracted from original patch image. The boundary of the mask is sometimes obvious.

$$\epsilon_0 = model\_predict(x_t, t) \tag{4}$$

$$x_0 = \frac{x_t - \sqrt{1-\bar{\alpha}_t}\epsilon_0}{\sqrt{\bar{\alpha}_t}} \tag{5}$$

$$x_{t-1} = \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}x_0 + \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}x_t + \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t \tag{6}$$

We conducted experiments that by training the cropped regions from defective images, the diffusion models can learn to generate useful samples (as is shown in Figure2). The Table1 shows the performance improvements by introducing generated samples while using a small number of real defect samples.

However, mura images and defects are variable in size, and the cropped region cannot be placed into the original region because of the mass changes brought by the randomness of the probabilistic model. This usually makes it impossible to train the model in a more flexible way for those images with high resolutions for manufacturing products. Also, the locations of the generated defects are unknown to us, which still requires additional labeling work. We need to measure information including defect locations, sizes, and visibility for the products.

3.2. DDIMs

DDIMs (denoising diffusion implicit models) has same training procedure as DDPMs but greatly reduces the sampling time by introducing a non-Markovian process [40]. By controlling τ (which decides the sampling interval across time steps), and η, we get σ (which interpolates between the deterministic DDIM and the stochastic DDPM)

$$\sigma_t = \eta \cdot \sqrt{\frac{1 - \bar{\alpha}_{\tau_{t-1}}}{1 - \bar{\alpha}_{\tau_t}}} \cdot \sqrt{1 - \frac{\bar{\alpha}_{\tau_t}}{\bar{\alpha}_{\tau_{t-1}}}} \tag{7}$$

Then by predicting the noise and the initial input:

$$\epsilon_0 = \epsilon^{(t)}(x_t) \tag{8}$$

$$\hat{x}_0 = \frac{x_t - \sqrt{1 - \bar{\alpha}_t}\epsilon_0}{\sqrt{\bar{\alpha}_t}} \tag{9}$$

The sampling procedure can be expressed as:

$$x_{t-1} = \sqrt{\bar{\alpha}_{t-1}}\hat{x}_0 + \underbrace{\sqrt{1 - \bar{\alpha}_{t-1} - \sigma_t^2} \cdot \epsilon_0}_{direction pointing\ to\ x_t} + \underbrace{\sigma_t \epsilon_t}_{random\ noise} \tag{10}$$

We found that setting a long sampling interval cannot provide high quality results. To make the balance between time and image quality, when time T is close to zero, we use DDPM instead of DDIM.

3.3. Defect Generation based on Conditional Diffusion

Inspired by the inpainting approaches [34], we apply masks and the diffusion process on the defect regions only. During the forward pass, only the masked region is changed by adding gaussian noise step by step. And in the backward pass, all pixels from x_t are fed into the network to predict the noise, but the calculation of loss only involves the masked regions.

With this approach, it's possible to generate defects in the regions we specified. However, the boundary can be visible (shown in Figure3) even in the training data when we try other regions that are not trained with masks. This is because defect samples from the products are quite scarce and defect areas are normally not large, which greatly reduces the actual available number of training data. The generation task, in other words, can be divided into two sub-tasks: background generation and defect generation. So in addition to the previous work, we add masks on a random region to try to let the model learn to reconstruct the backgrounds and introduce additional information as conditions to the input of the model, with mask regions filled with 1 representing defect area to generate and 0 representing the background area to learn.

Although the generation ability is further improved with conditional input and masks, different from other kinds of products, an image of a display where mura defects may exist is usually filled with random noises, which to some extent follows Gaussian distribution. Since the gap between high-density and low-density pixels of a pattern is not large, the randomness of the noise makes it harder for the network to learn the data distribution due to normal loss settings. Thus, it is found that sometimes for complex background patterns observable with human eyes, the diffusion model may fail or need much more training time to generate an undistinguishable defect region. However, with traditional image processing approaches to detect many kinds of defects, we usually apply Gaussian blur to the image first. So a 3x3 Gaussian kernel is carefully added to the original image to make sure the defective features are not changed before being fed into the model.

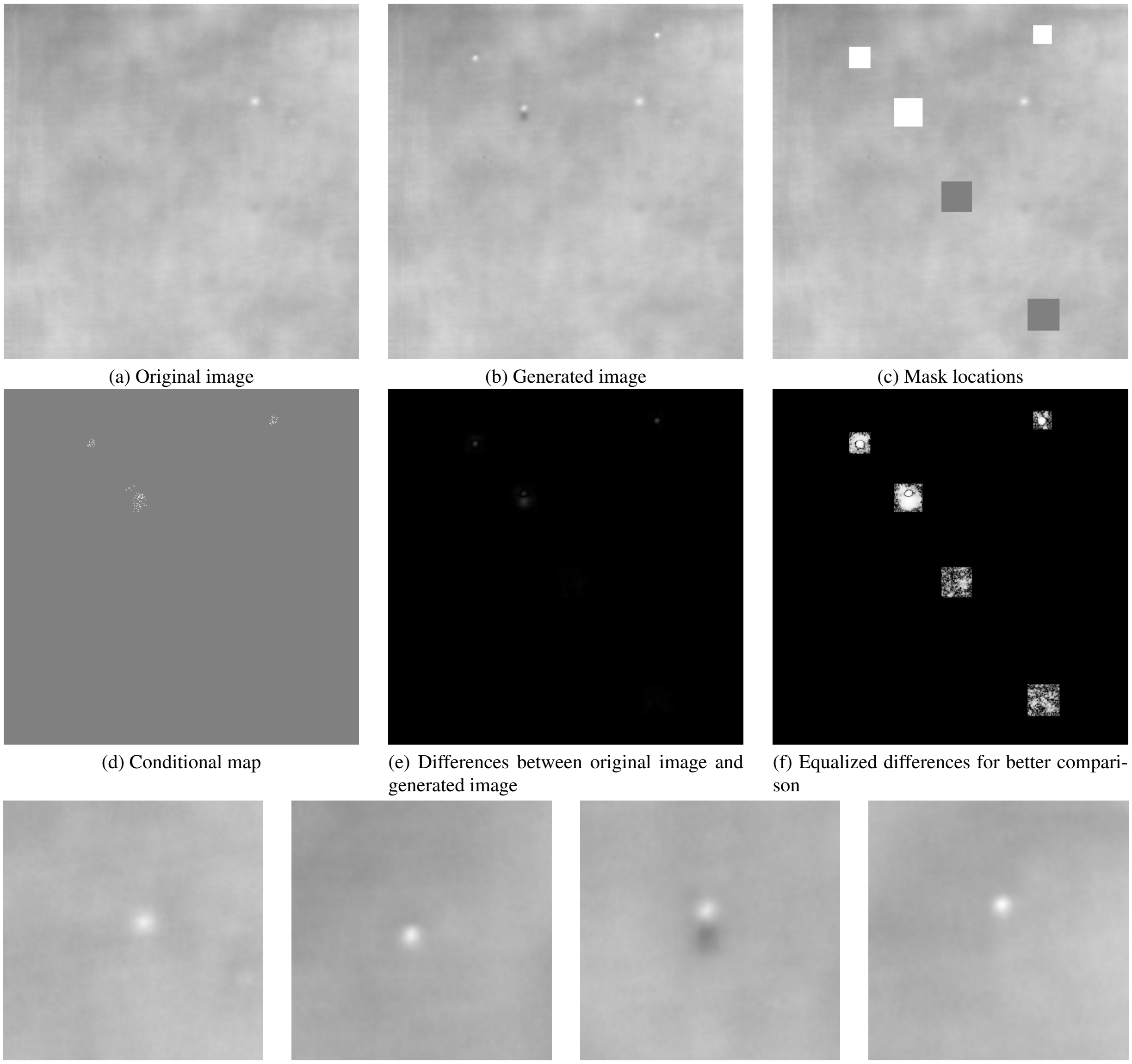
3.4. Feature Reservation

Different from inpainting tasks on normal images, defects usually interrupt the continuity of the background patterns. The process of learning to generate an area lacking enough context information may be more unpredictable and harder because of loss settings. We try to use Canny together with normalization techniques to keep the missing features for the mask regions that are used to generate defects. However, with this approach, additional dependencies on existing defective data are required for testing and massive defect generation stages because we have to simulate existing edge features. To relieve this, we use dropout in the following step to keep much fewer features about the defect regions to decouple the dependencies.

For training, edges and positions with high normalization values are set to 1, being concatenated with the inputs. At the same time, those values are also stored as images for further defect generation. And during the testing stage, similar settings are applied, with a conditional map containing edge and position information within the mask region. However, to generate more defect images, we can randomly select the stored feature images collected in the training stage and paste them onto the masked regions as conditional maps, as illustrated in Algorithm1.

4. Experimental Results

Our experiments include two parts. The first part, a micro test to ensure that an indistinguishable generated defect can actually improve deep learning performance, uses around 200 different samples to simply discover the generation ability of DDPM. And the second part, we test the performance of the location-specified defect generation.



(a) Original image

(b) Generated image

(c) Mask locations

(d) Conditional map

(e) Differences between original image and generated image

(f) Equalized differences for better comparison

(g) Real defect 1

(h) Fake defect 1

(i) Fake defect 2

(j) Fake defect 3

Figure 4. A set of images illustrating our methods and outcomes. In Figure (c), the white boxes on the mask image indicate the regions we want to generate defects, and the black boxes indicate the regions we want to generate normal background. The sizes and locations of black and white boxes are random. (d) Shows the condition map, which is loaded from the disk and saved during the training process, as input to the diffusion model, after being dropped out at a random rate. (e) compares the differences between the original image and generated image, and (f) provides another view by using histogram equalization to (e) to present all the different regions. (g) to (h) are the real and generated defect regions.

4.1. Dataset

Our dataset contains four kinds of spot mura and a kind of shapeless mura, splitted into training and testing datasets respectively. Images from different products are cropped to small patches (640x640) and then resized to a smaller resolution (512x512) as input to the diffusion models.

The first experiment uses around 200 different defect patches for DDPM training. To test the generation ability, 15 of the patches are used for supervised training using Yolov5 while another group has 47 more generated samples, and additional 10 patches are used for performance testing. The second experiment uses around 700 different patches for training and 200 patches for testing.

Algorithm 1 The generation of condition map

```
Require: images, position labels
Output: $condition\_map$
while Generation process not finished do
    $img \leftarrow GaussionBlur(img)$ (optional)
    $bmsk \leftarrow$ Get random background mask
    $fmask \leftarrow$ Get foreground mask
    $cond \leftarrow img(1 - fmsk \cup bmsk) + fmsk$
    $dropou\_rate = random(0.4, 0.9)$
    $\mathcal{R} \leftarrow$ non-zero regions of cond
    for each $r \in \mathcal{R}$ do
        $c \leftarrow$ Get Canny map from $r$
        $n \leftarrow$ Perform normalization on $r$
        $n \leftarrow$ Update $n$ according to threshold settings
        $\hat{r} \leftarrow n \cup c$
        $cond_{r \in \mathcal{R}} \leftarrow \hat{r}$
    end for
    $condition\_map \leftarrow dropout(cond, dropout\_rate)$
end while
```

4.2. Experiment Settings

The first experiment, with no region specified, is based on [25]. We adopt most of the original settings but increase the resolution to 512x512. And the second experiment, with the location specified, is based on [18]. Along with additional augmentation approaches such as random rotation and flipping, we modified the mask, and condition map generation scheme as illustrated in the last section.

4.3. Quality Matrix

Table1 shows the performance improvements brought by the generated defect samples with DDPM when the training dataset is very small, which indicates those fake defects can be valid inputs for training supervised deep learning models.

Table 1. Performace comparison using YoloV5 on different data

Case	Real data	Generated data	Real+generated data
Number of samples	15	47	62
Recall	0.27	0.45	0.55
Precision	0.21	0.31	1.0

Figure4 compares the real and generated mura defects, which are indistinguishable with human eyes. For comparing quality between different models, we perform quantitative evaluations by the following metrics:

Inception Score (IS) [36], measures how well a model captures the full ImageNet class distribution while still producing individual samples. One drawback of this metric [3] is that it does not reward covering the whole distribution or capturing diversity within a class, and models which memorize a small subset of the full dataset will still have high IS.

Fréchet Inception Distance (FID) [13], was proposed to better capture diversity than IS, which is more consistent with human judgment than Inception Score. FID provides a symmetric measure of the distance between two image distributions in the Inception-V3 latent space [43]. Recently, sFID [29] was proposed to replace spatial features with standard pooled features.

We use **FID** as our default metric for overall quality comparisons as it captures both diversity and fidelity and has been adopted by generative modeling work [14, 20].

Table 2. Evaluation and comparisons on the generated data

Case	FID score
Conditioned diffusion with Gaussian Blur	17.34
Feature aware diffusion	7.23

5. Conclusions and Future Work

5.1. Conclusions

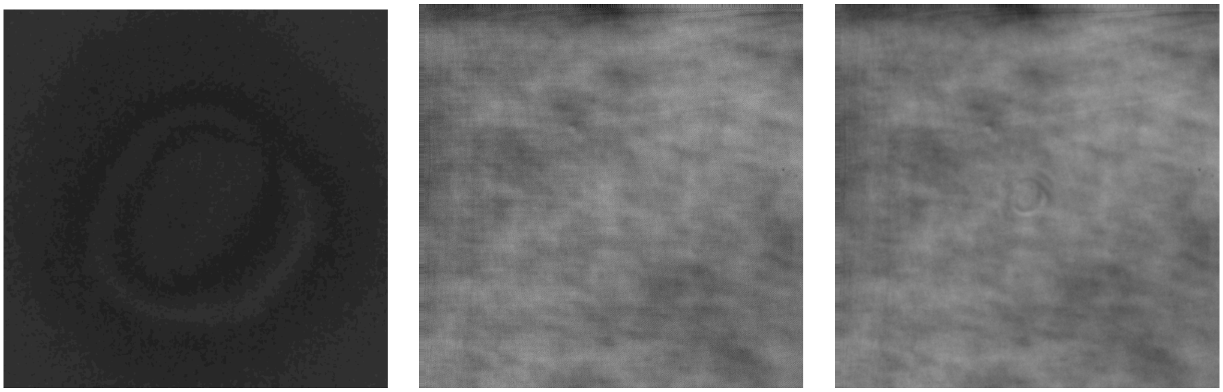
In this paper, we presented a novel approach for assigned MURA generation using a conditional diffusion model. The proposed approach involves training a diffusion model on the limited MURA dataset from the factory with professional annotation, which learns to generate realistic MURA images. The generated MURA images not only contain the assigned defect class but also the locations of MURA. We evaluated the approach to the detection task using the MURA dataset w/o generating MURA images. Experimental results show that it is able to improve MURA detection accuracy.

5.2. Future work

Common defect detection methods based on unsupervised learning learn to reconstruct the background, which can work well when the backgrounds are simple. However, because of the complex production environments, noises and real defects can be hard to be distinguished. It should be observed that many defects appearing on different products share the same features, which provides a possibility for the network to reconstruct them.

However, learning to reconstruct defects involves learning to reconstruct backgrounds as well. Our experiments show that with rough skeletons reserved for the unseen region, the network is well capable of reasoning for both the backgrounds and defects.

Consequently, future research will mainly focus on defect extraction, translation, and generation across a wider variety of products, as is shown in Figure5.



(a) Real extracted defect from other another product (only defect patch shown for data confidential)

(b) Real image

(c) Generated sample with defect

Figure 5. Example of defect extraction and generation

Defect extraction By reconstructing the region of defects, it's possible to extract the defects with acceptable noises(Figure5(a)).

Defect translation Translate defects from one product to another product in any specified location (Figure5(c)).

Defect generation With the help of diffusion models, it's feasible to learn the distribution of defects separately and reconstruct a wider range of defects.

References

[1] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 214–223. PMLR, 06–11 Aug 2017. 2

[2] Jacob Austin, Daniel D. Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. Structured denoising diffusion models in discrete state-spaces. *CoRR*, abs/2107.03006, 2021. 2

[3] Shane Barratt and Rishi Sharma. A note on the inception score, 2018. 6

[4] Ruojin Cai, Guandao Yang, Hadar Averbuch-Elor, Zekun Hao, Serge J. Belongie, Noah Snavely, and Bharath Hariharan. Learning gradient fields for shape generation. *CoRR*, abs/2008.06520, 2020. 2

[5] Nanxin Chen, Yu Zhang, Heiga Zen, Ron J. Weiss, Mohammad Norouzi, and William Chan. Wavegrad: Estimating gradients for waveform generation, 2020. 2

[6] Nanxin Chen, Yu Zhang, Heiga Zen, Ron J. Weiss, Mohammad Norouzi, Najim Dehak, and William Chan. Wavegrad 2: Iterative refinement for text-to-speech synthesis, 2021. 2

[7] Florinel-Alin Croitoru, Vlad Hondru, Radu Tudor Ionescu, and Mubarak Shah. Diffusion models in vision: A survey. *arXiv preprint arXiv:2209.04747*, 2022. 2

[8] Prafulla Dhariwal and Alex Nichol. Diffusion models beat gans on image synthesis. *CoRR*, abs/2105.05233, 2021. 2

[9] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp. 2016. 2

[10] Shu-Kai S. Fan and Yu-Chiang Chuang. Automatic detection of mura defect in tft-lcd based on regression diagnostics. *Pattern Recognit. Lett.*, 31:2397–2404, 2010. 1

[11] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014. 2

[12] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. 2

[13] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, Günter Klambauer, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a nash equilibrium. *CoRR*, abs/1706.08500, 2017. 6

[14] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 6840–6851. Curran Associates, Inc., 2020. 2, 6

[15] Jonathan Ho, Chitwan Saharia, William Chan, David J. Fleet, Mohammad Norouzi, and Tim Salimans. Cascaded diffusion models for high fidelity image generation. *CoRR*, abs/2106.15282, 2021. 2

[16] Emiel Hoogeboom, Didrik Nielsen, Priyank Jaini, Patrick Forre, and Max Welling. Argmax flows and multinomial diffusion: Learning categorical distributions, 2021. 2

[17] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. 2017. 2

[18] Janspiry. Palette-image-to-image-diffusion-models. 6

[19] Glenn Jocher. ultralytics/yolov5: v3.1 - Bug Fixes and Performance Improvements. Zenodo, Oct. 2020. 2

[20] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. *CoRR*, abs/1812.04948, 2018. 6

[21] Durk P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. 2

[22] Diederik P. Kingma, Tim Salimans, Ben Poole, and Jonathan Ho. Variational diffusion models. *CoRR*, abs/2107.00630, 2021. 2

[23] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In Yoshua Bengio and Yann LeCun, editors, *ICLR*, 2014. 2

[24] Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro. Diffwave: A versatile diffusion model for audio synthesis, 2020. 2

[25] lucidrains. denoising-diffusion-pytorch. 6

[26] Chenlin Meng, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. Sdedit: Image synthesis and editing with stochastic differential equations. *CoRR*, abs/2108.01073, 2021. 2

[27] Luke Metz, Ben Poole, David Pfau, and Jascha Sohl-Dickstein. Unrolled generative adversarial networks. volume abs/1611.02163, 2016. 2

[28] Wuyi Ming, Shengfei Zhang, Xuewen Liu, Kun Liu, Jie Yuan, Zhuobin Xie, Peiyan Sun, and Xudong Guo. Survey of mura defect detection in liquid crystal displays based on machine vision, 2021. 1, 2

[29] Charlie Nash, Jacob Menick, Sander Dieleman, and Peter W. Battaglia. Generating images with sparse representations. *CoRR*, abs/2103.03841, 2021. 6

[30] Alex Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. *CoRR*, abs/2102.09672, 2021. 3

[31] Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Lukasz Kaiser, Noam Shazeer, Alexander Ku, and Dustin Tran. Image transformer. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 4055–4064. PMLR, 10–15 Jul 2018. 2

[32] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks, 2015. 2

[33] Suman V. Ravuri and Oriol Vinyals. Classification accuracy score for conditional generative models. volume abs/1905.10887, 2019. 2

[34] Chitwan Saharia, William Chan, Huiwen Chang, Chris Lee, Jonathan Ho, Tim Salimans, David Fleet, and Mohammad Norouzi. Palette: Image-to-image diffusion models. In *ACM SIGGRAPH 2022 Conference Proceedings*, pages 1–10, 2022. 2, 4

[35] Chitwan Saharia, Jonathan Ho, William Chan, Tim Salimans, David J. Fleet, and Mohammad Norouzi. Image super-resolution via iterative refinement, 2021. 2

[36] Tim Salimans, Ian J. Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *CoRR*, abs/1606.03498, 2016. 6

[37] Hiroshi Sasaki, Chris G. Willcocks, and Toby P. Breckon. UNIT-DDPM: unpaired image translation with denoising diffusion probabilistic models. *CoRR*, abs/2104.05358, 2021. 2

[38] Abhishek Sinha, Jiaming Song, Chenlin Meng, and Stefano Ermon. D2C: diffusion-denoising models for few-shot conditional generation. *CoRR*, abs/2106.06819, 2021. 2

[39] Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. *CoRR*, abs/1503.03585, 2015. 2

[40] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *CoRR*, abs/2010.02502, 2020. 3, 4

[41] Yang Song and Stefano Ermon. Improved techniques for training score-based generative models. *CoRR*, abs/2006.09011, 2020. 2

[42] Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *CoRR*, abs/2011.13456, 2020. 2

[43] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. *CoRR*, abs/1512.00567, 2015. 6

[44] Arash Vahdat and Jan Kautz. Nvae: A deep hierarchical variational autoencoder, 2020. 2

[45] Arash Vahdat, Karsten Kreis, and Jan Kautz. Score-based generative modeling in latent space, 2021. 2

[46] Aaron van den Oord, Nal Kalchbrenner, Lasse Espeholt, koray kavukcuoglu, Oriol Vinyals, and Alex Graves. Conditional image generation with pixelcnn decoders. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. 2

[47] Min Zhao, Fan Bao, Chongxuan Li, and Jun Zhu. Egsde: Unpaired image-to-image translation via energy-guided stochastic differential equations, 2022. 2