2023 IEEE International Conference on Industry 4.0, Artificial Intelligence, and Communications Technology (IAICT) | 979-8-3503-1363-5/23/$31.00 ©2023 IEEE | DOI: 10.1109/IAICT59002.2023.10205662

ResUnet-GAN with Dynamic Memory for Mura Defect Detection

Chia-Yu Lin*, Pin-Fan Lin†, Wei-Kuang Chung†, and Yu-Hsien Lee†
* *Department of Computer Science and Information Engineering, National Central University, Taoyuan, Taiwan*
† *Department of Computer Science and Engineering, Yuan Ze University, Taoyuan, Taiwan*
Corresponding Author's E-mail: sallylin0121@ncu.edu.tw

***Abstract*—"Mura" is a phenomenon in which panels have uneven display defects, irregular shapes, and different sizes. It is impossible to produce perfect panels on production lines, so panel inspection is necessary to differentiate between "light Mura" and "serious Mura" manually. The performance of conventional defect detection models for Mura detection is worse since they only differentiate between "normal" and "abnormal" samples. To reduce human cost and increase the accuracy of Mura detection, we propose a "ResUnet-GAN with Dynamic Memory Model," an unsupervised anomaly detection method based on a Generative Adversarial Network (GAN) with a memory module to distinguish panel defects. In the dynamic memory, we designed a dynamic feature filtering (DFF) method to choose important features of images, enhancing the ability to recognize light Mura features of the ResUnet-GAN. The proposed model can achieve an Area Under Curve (AUC) of approximately 0.8 for accurate Mura detection. The mechanism of this paper is novel, and the result contributes to practical application.**

***Index Terms*—Mura detection, ResNet, U-Net, GAN, dynamic feature filtering**

I. Introduction

In recent years, Mura defect detection has played an essential role in panel manufacturing before products are shipped. The detection must be inspected in detail to ensure the panels can be used. We hope that AI can help the factory reduce the human cost of detection.

Generally speaking, two Mura defect detection stations will be set up in the production line. The first station usually uses lots of human resources to repair the defects of all panels. The panels with light imperfections can be fixed to normal, and those with more severe and unbalanced luminosity panels should be sent to the next station. With the development of AI, defect detection is no longer inspected by people. However, in the past few years, detection can only be divided into "normal" and "abnormal," and it takes a lot of human costs to check whether the panel is standard. To reduce the work of the first station, we hope the panels can be divided into "light Mura that can be fixed," as shown in Fig. 1 and "serious Mura," such as the butterfly, as shown in Fig. 2 before the panels enter the first station to reduce the human cost.

However, in most cases, the number of serious Mura samples accounts for very few. So it is difficult to balance the numbers of light Mura and serious Mura, which leads to the supervised learning model being hard to classify because of data unbalancing. It will cause the accuracy of the model to be significantly decreased.

Fig. 1. Light Mura.

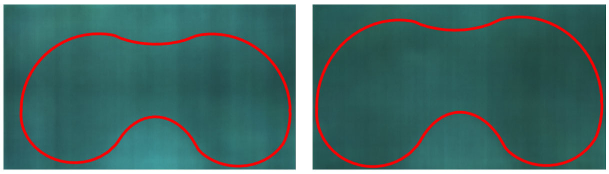

Fig. 2. The serious Mura with butterfly.

Anomaly detection has been rapidly developed under deep learning in response to this situation. Due to Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), and Long Short-Term Memory (LSTM) have already achieved specific results in technology.

We designed the model using few and unlabeled data to identify "light Mura" and "serious Mura." In this study, we take light Mura as standard data to train the model. Due to Mura being too small, it is hard to detect, so the panel's image will be preprocessed first by using a sliding crop to crop the images into several segments, which will be our new data. Finally, training an unsupervised GAN with a dynamic memory module based on ResUnet (Mem-ResUnet-GAN) and the Area Under Curve (AUC) can go up to almost 0.8.

II. Related Work

The traditional defect detection method used classification models to classify different defects [1]. The network structure was divided into a de-noising network and a classification network. However, when the abnormal data was insufficient, the classification model could not be trained with high quality. Therefore, unsupervised learning was proposed to take normal data for model training.

[2] proposed GANomaly based on AnoGAN and Efficient-GAN-Anomaly. When training the normal data in the autoencoder, it encoded and decoded abnormal data that had never been seen before. The potential space difference was often significant after the encoding processes. When the difference was larger than the threshold, it was considered an abnormal sample. Since the details and the texture features

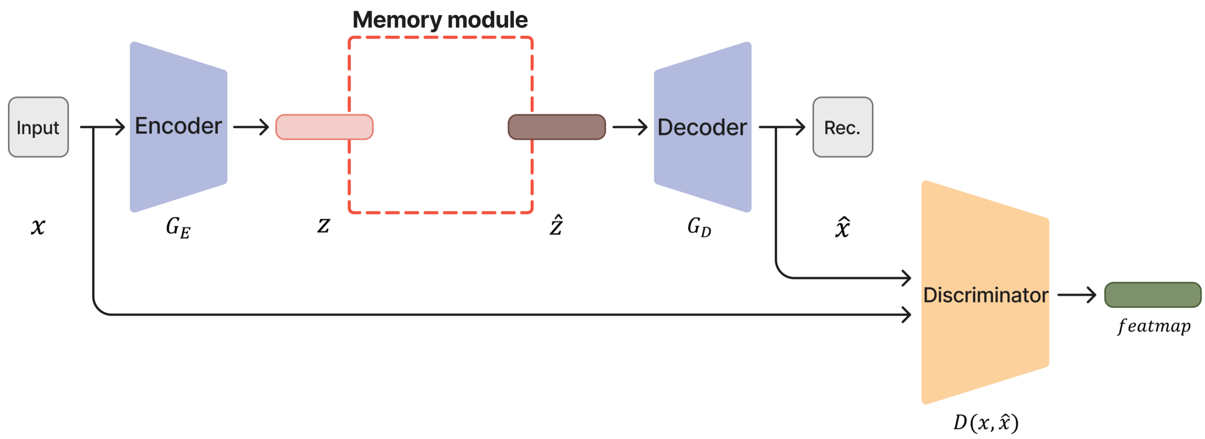

Fig. 3. The architecture of Mem-ResUnet-GAN.

of the image were mainly located at the bottom layer of the CNN, and the advanced abstract features were located at the top layer. The latent vector did not contain enough features, so the reconstruction easily lost details. Therefore, Skip-GANomaly [2] was inspired by U-Net and adopted skip-connection to connect each down-sampling layer in the encoder network to its corresponding up-sampling decoder layer.

[3] proposed a memory-augmented autoencoder (MemAE) to improve the reconstruction ability of AE by adding the memory module to store the prototypical elements of the normal data and utilize a differentiable hard shrinkage operator to induce sparsity of the memory addressing weights, which let the memory items be close to the query in the feature space. However, the threshold of the hard shrinkage operator was set to be fixed. It could reach a good result, but it could not be better. So, we combined ResUnet-GAN with a memory module and used dynamic feature filtering (DFF) that could solve this problem.

III. The Algorithms

This project intends to design a "Mem-ResUnet-GAN for Mura defect detection" to identify light Mura and serious Mura. In the architecture shown in Fig. 3, we use the "data with a light Mura" to treat such data as standard data and input it into the GAN with a memory module to train a model of a healthy panel. During testing, since the training data is all standard images, prototypical features are stored in the memory. Therefore, when the input is a severely flawed panel, the reconstructed image will still reconstruct a near-healthy image. It will cause the reconstructed and original images to make a big difference and help distinguish.

A. Image Preprocessing

According to the previous experiments, we found that the size of the Mura is about 64 x 64, so we used the sliding crop to crop the image from 512 x 512 into multiple 64 x 64 segments to be our new dataset, as shown in Fig. 4. In this way, it can find the Mura in the panel precisely.

B. Mem-ResUnet-GAN for Mura Defect Detection

Mem-ResUnet-GAN is constructed based on ResUnet-GAN [4]. The preprocessed light Mura image is the training data that is constructed based on ResUnet-GAN.

1) Encoder: It uses the pre-trained model ResNet50 model to extract the features. The advantage of the ResNet50 is that it consists largely of the network layer. It has a more vital ability to extract the features and can get a better feature vector to help us get a reconstructed image closer to the original.

2) Memory: We design memory proposed by [3]. In our model, the memory $\mathsf{M} \in \mathbb{R}^{N \times K}$ where N stands for the size of memory, and K stands for the dimension of the latent vector z, is design to store the latent features vector $\mathsf{z} \in \mathbb{R}^{K}$ extracted by the encoder as shown in Fig. 3. And the weight w_i of the corresponding features to be a metric to determine which feature helps construct the image closer to the original image. We compute each w_i via a softmax operation as follows:

$$w_i = \frac{\exp(d(\mathsf{z}, \mathsf{m}_i))}{\sum_{j=1}^{N} \exp(d(\mathsf{z}, \mathsf{m}_j))} \tag{1}$$

where $d(\cdot,\cdot)$ denotes a similarity measurement, refer to the [3], we define $d(\cdot,\cdot)$ as cosine similarity:

$$d(\mathsf{z}, \mathsf{m}_i) = \frac{\mathsf{z}\mathsf{m}_i^T}{\|\mathsf{z}\|\|\mathsf{m}_i\|} \tag{2}$$

In the training phase, to restrict the use of memory items for reconstruction, a method for dynamic memory (See Section C) to effectively utilize the memory is proposed so that model can use the most representative standard features.

3) Decoder: The decoder adopts the skip-connection method so that each down-sampling layer in the encoder network is connected to its corresponding up-sampling decoder layer. The use of this skip-connection allows for excellent and stable reconstructions in both high-dimensional image space and low-dimensional latent vector space encoding to improve overall model resolution accuracy.

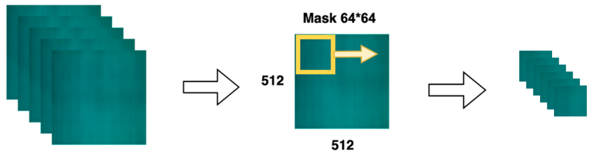

Fig. 4. Image preprocessing.

4) Discriminator Network: As an adversarial network, the primary method to discriminate between the original and the reconstructed image is conducting confrontation and obtaining the anomaly score to judge whether the image is defective. It uses the DCGAN model [5] and the structure of the Depth Separable Convolution [6] for optimization, reducing the number of parameters of the network model.

C. Dynamic Memory

Using the limited number of standard features for reconstruction helps the reconstructed image have significant differences from the original. The reason is that we use light Mura as normal data for training; there may still be some features with flaws. If we take all of the memory items to reconstruct, it will cause the tiny defective features to be restored.

Therefore, we proposed a dynamic feature filtering (DFF) method that can utilize memory items effectively. DFF sets a threshold to filter features based on each image. Because every feature has its corresponding weight, we will select how many percentages of features will be used. It can filter the features that are not important, helping us to distinguish the panel with light Mura or serious Mura. After doing the filtering, we will take the features that store in memory items as shown in Fig. 6. The weight $\widehat{w}_i$ calculated after filtering as shown in Eq. (3)

$$\widehat{w}_i = \frac{\max(w_i - p, 0) \cdot w_i}{|w_i - p| + \varepsilon} \tag{3}$$

where $\max(\cdot, 0)$ is called the ReLU activation, p is the first few percentiles features we take in this image, and ε is a very small positive scalar.

IV. Experiment

In this section, we demonstrate the effectiveness of our proposed method. We use dynamic memory and set different percentages threshold to compare whether we take the few features that help reconstruct Mem-ResUnet-GAN.

A. Data

We are using high-resolution datasets to do experiments from the manufacturer. And we took 4835 standard images to be a training dataset, 541 standard images, and 143 serious Mura images to do the testing. The original size of high-resolution datasets is 1920 x 1080, and for the convenience of training, we resize the original images into 512 x 512. Then, using sliding crop to turn images into multiple sizes of 64 x 64 small pictures, randomly selecting 64 images for training.

TABLE I
The accruracy of Mem-ResUnet-GAN with different threshold

Threshold	Precision	Recall	TNR	AUC
20%	0.496	0.496	0.841	0.696
10%	0.609	0.483	0.902	0.734
5%	0.567	0.562	0.864	0.769

TABLE II
Accuracy comparison with different models

Model	Precision	Recall	TNR	AUC
Skip-GANomaly	0.036	0.852	0.163	0.480
ResUnet-GAN	0.546	0.660	0.657	0.648
Mem-ResUnet-GAN w/ DFF	0.567	0.562	0.864	0.769

B. Model Parameter

In the part of the model, we used the Mem-ResUnet-GAN model. Setting batch size = 64, stride = 32 for sliding crop and Epoch = 400, $lr = 1e^{-4}$, and the memory size = 2000.

C. Evaluation Index (AUC)

AUC represents the area under the Receiver Operator Characteristic (ROC) curve, a standard statistic that can stand for the predictive ability of a classifier. And ROC curve takes False Positive Rate (FPR) as the X-axis and True Positive Rate (TPR) as the Y-axis, which means the relative relationship between TPR and FPR of a classifier.

We are taking anomaly scores to evaluate whether the images are defective or not. The method is referred to [4]. In the testing phase, we do the same processing to the testing datasets. Due to the stride setting 32, it will generate 256 images. We take those small images to count anomaly scores and take the highest score as a representation.

D. Experiments

We take the top 20%, the top 10%, and the top 5%, respectively. The experimental results are shown in Table I. It can be known that the smaller the threshold is, the more correlation memory items are taken out, and the AUC is also higher.

E. Comparison

We also compare different models to verify whether our model is more prominent. This section presents the experimental results of Skip-GANomaly and ResUnet-GAN without image preprocessing.

In Table II, we can see that the Skip-GANomaly only gets AUC 0.48. Although its recall has 0.852, it can't distinguish the standard images. However, ResUnet-GAN can go to 0.648; its powerful image feature extraction ability helps us restore the image better when reconstructing it.

We also did the Mem-ResUnet-GAN experiment. It can be seen that after adding the memory module, the AUC has been significantly improved by 12.1%. It can be seen that the memory module can help to better distinguish between light Mura and serious Mura.

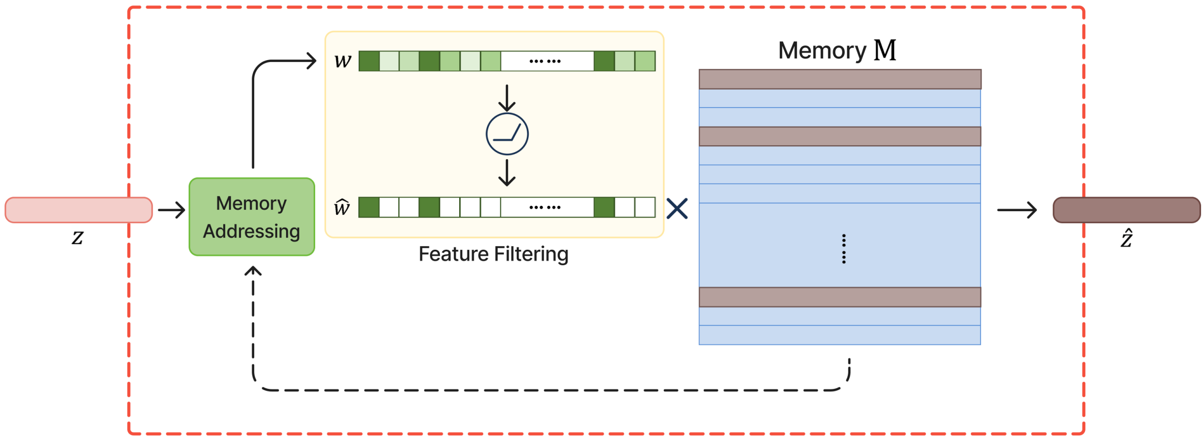

Fig. 5. Memory module.

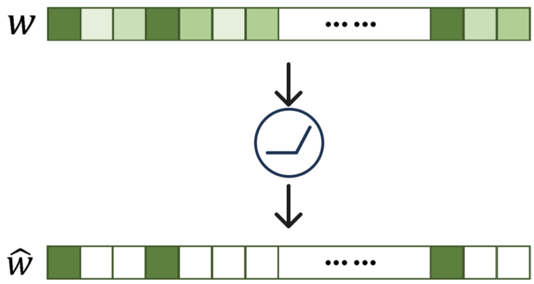

Fig. 6. Feature filtering.

TABLE III
THE EFFECTS OF MEMORY IN MEM-RESUNET-GAN

Model	Precision	Recall	TNR	AUC
Mem-ResUnet-GAN	0.514	0.576	0.538	0.738
Mem-ResUnet-GAN w/ DFF	0.567	0.562	0.864	0.769

In addition, we also compared Mem-ResUnet-GAN with DFF or not. From Table III, it can be seen that the AUC when DFF is not used is only about 0.738, but after adding DFF, the overall AUC is increased to 0.769, an increase of 3.1%.

V. CONCLUSION

We proposed a Mem-ResUnet-GAN with dynamic memory to accurately distinguish "light Mura" and "serious Mura." The sliding crop pre-process method and GAN with the memory module were designed to increase the model performance. In the encoder part, we added the memory module to store the prototypical latent features vector and used DFF to keep the most important features to help the decoder reconstruct the image. The AUC of the proposed model can reach almost 0.8. The proposed ResUnet-GAN with dynamic memory cannot only filter the important features of light Mura but intensively improve the performance of Mura inspection.

ACKNOWLEDGEMENTS

This work is jointly sponsored by AUO Corporation, AUO · NYCU Joint Research and Development Center, National Central University, Yuan Ze University, and National Science and Technology Council (NSTC) under the project NSTC 111-2622-8-A49-023 and NSTC 110-2222-E-008-008-MY3.

REFERENCES

[1] Hsueh-Ping Lu and Chao-Ton Su, "Cnns combined with a conditional gan for mura defect classification in tft-lcds," *IEEE Transactions on Semiconductor Manufacturing*, vol. 34, no. 1, pp. 25–33, 2021.
[2] Samet Akcay, Amir Atapour-Abarghouei, and Toby P Breckon, "Ganomaly: Semi-supervised anomaly detection via adversarial training," in *Computer Vision–ACCV 2018: 14th Asian Conference on Computer Vision, Perth, Australia, December 2–6, 2018, Revised Selected Papers, Part III 14*. Springer, 2019, pp. 622–637.
[3] Dong Gong, Lingqiao Liu, Vuong Le, Budhaditya Saha, Moussa Reda Mansour, Svetha Venkatesh, and Anton van den Hengel, "Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 1705–1714.
[4] Shubin Song, Kecheng Yang, Anni Wang, Shengsen Zhang, and Min Xia, "A mura detection model based on unsupervised adversarial learning," *IEEE Access*, vol. 9, pp. 49920–49928, 2021.
[5] Alec Radford, Luke Metz, and Soumith Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015.
[6] Lukasz Kaiser, Aidan N Gomez, and Francois Chollet, "Depthwise separable convolutions for neural machine translation," *arXiv preprint arXiv:1706.03059*, 2017.