Deep Learning Final Project

His-Ling Chen
Institute of Computer and Communication Engineering
Department of Electrical Engineering
National Cheng Kung University, Tainan, Taiwan
nn6114027@gs.ncku.edu.tw

Wei-Hai Huang
Institute of Statistics
Department of Statistics
National Cheng Kung University, Tainan, Taiwan
robert20000831@gmail.com

Yi-No Tseng
Institute of Data Science
Department of Statistics
National Cheng Kung University, Tainan, Taiwan
re6111032@gs.ncku.edu.tw

Chun-Hsien Yang
Institute of Data Science
Department of Statistics
National Cheng Kung University, Tainan, Taiwan
re6111040@gs.ncku.edu.tw

Abstract **- In this project, we design a new architecture SemanticDiffGAN for anomalous image semantic segmentation using diffusion model combined with GAN model, and our proposed approach is experimentally proven to be a feasible direction in multiple anomaly detection tasks.**

GitHub Link **- https://github.com/butterfly2012010/A-Fusion-of-Diffusion-and-GAN**

Index Terms – Diffusion model, GAN, Semantic segmentation, Anomaly detection, Weakly supervised learning.

I. INTRODUCTION

A. Research background

Semantic segmentation tasks often require meticulous pixel-level annotations, which can be time-consuming and labor-intensive. Manual labeling is prone to errors, leading to unstable results. To overcome these challenges, weakly supervised learning has emerged as an alternative. Instead of pixel-level annotations, weakly supervised learning utilizes image-level annotations, reducing the annotation workload and speeding up the labeling process while still achieving meaningful segmentation results.

Anomaly detection has become increasingly important in various fields, particularly in medicine and industry. In the medical domain, anomaly detection aids professionals in real-time analysis, facilitating the identification of abnormalities in medical images such as tumors or lesions for early diagnosis and treatment planning. In the industrial domain, anomaly detection can monitor equipment, detecting malfunctions or deviations from normal operations to prevent potential failures and ensure efficiency.

Developing robust anomaly detection algorithms is crucial for enhancing safety, optimizing resource utilization, and reducing downtime in critical applications. Leveraging advanced techniques like deep learning and computer vision, anomaly detection systems can learn patterns and anomalies from large-scale datasets, enabling efficient and accurate identification of abnormal instances. Integrating weakly supervised learning with anomaly detection allows researchers to explore innovative approaches, leveraging limited annotations to improve detection performance. This integration leads to scalable and cost-effective anomaly detection systems adaptable to various domains, contributing to better decision-making processes.

B. Motivation

Recent research has focused on image generation using different approaches, including Generative Adversarial Networks (GANs) and Autoencoders. However, both methods have limitations and face challenges. GANs often suffer from training difficulties and suboptimal results due to instability issues like mode collapse and convergence problems. Autoencoders struggle to generate diverse and high-quality images due to constraints imposed by their architecture.

In contrast, diffusion models have emerged as a promising alternative for image generation. They offer several advantages over GANs and Autoencoders. Diffusion models are known for their easy training process, which is more stable and less prone to issues encountered in GANs. They also provide better interpretability and understanding, as the generative process follows a step-by-step diffusion process that incrementally refines the image by adding noise. This allows for a clearer understanding of the generation process compared to complex architectures like GANs.

Furthermore, diffusion models excel in generating high-quality and diverse images. Their iterative nature provides fine-grained control, resulting in visually appealing and realistic samples with a wide range of variations. The combination of good training properties, interpretability, and high-quality diversity makes diffusion models a promising direction in image generation research. Researchers are actively working on improving diffusion models to overcome the limitations of previous approaches.

By leveraging the strengths of diffusion models, it becomes possible to generate high-quality images that exhibit both creativity and fidelity. This opens up exciting possibilities in various applications such as computer graphics, art, and data augmentation for machine learning tasks.

C. Contribution

- We propose a new solution to the anomalous image semantic segmentation task

- We use GAN to replace the artificially designed noise to extend the application value of the model

II. RELATED WORK

In this section we review three studies that are most relevant to our approach.

A. Diffusion model

The diffusion model is a type of generative model consisting of two processes: the forward process and the reverse process (as shown in the figure above). In the forward process, pre-designed noise is added at each time step, aiming to approximate the image distribution to N(0,1) at the target time point (as described by the equation below).

$$\mathrm{q}(\mathrm{x_t}|\mathrm{x_{t-1}}) = N\left(\mathrm{x_t}\,;\sqrt{1-\beta_t}\mathrm{x_{t-1}},\beta_t\mathrm{I}\right) \text{ 錯誤! 尚未定義書籤。}, \quad (1)$$

where β_t is a coefficient.

The reverse process involves a deep learning model (U-net) learning the noise at each time step, with the goal of generating a clean and realistic output image (as described by the equation below).

$$\mathrm{p_\theta}(\mathrm{x_{t-1}}|\mathrm{x_t}) = N\left(\mathrm{x}_{t-1}\,;\,\mu_\theta(\mathrm{x}_t,t),\Sigma_\theta(\mathrm{x}_t,t)\right). \quad (2)$$

This formulation is further applied as follows:

$$\mathrm{p_\theta}\left(\mathrm{x_{t-1}}|\mathrm{x_{t,}x_{0,}}\right) = \exp\left(-\tfrac{1}{2}\left(\left(\tfrac{\alpha_t}{\beta_t}+\tfrac{1}{1-\bar{\alpha}_{t-1}}\right)\mathrm{x}_{t-1}^2 - \left(\tfrac{2\sqrt{\alpha_t}}{\beta_t}\mathrm{x}_t+\tfrac{2\sqrt{\bar{\alpha}_{t-1}}}{1-\bar{\alpha}_{t-1}}\mathrm{x}_0\right)\mathrm{x}_{t-1}+C(\mathrm{x}_t,\mathrm{x}_0)\right), \quad (3)$$

where α_t = 1-β_t.

Finally, the loss for each step is calculated as follows:

$$\nabla_\theta||\varepsilon-\varepsilon_\theta(\sqrt{\bar{\alpha}_t}\mathrm{x}_0+\sqrt{1-\bar{\alpha}_t\varepsilon}\,,\mathrm{t})||^2, \quad (4)$$

where $\varepsilon\sim N(0,I), \varepsilon_\theta$ represents the noise learned by the U-net.

B. Noise from different types

The author of this article used Simplex Noise instead of Gaussian Noise. Simplex Noise is an improved version based on Perlin Noise, with a time complexity of O(n^2) compared to Perlin Noise's O(2^n), which reduces computational burden in high-dimensional scenarios. For the two-dimensional case, the generation process of Perlin Noise is as follows: first, random gradients are sampled on a grid. For any given candidate point, the gradient and the dot product between the candidate point offset and the gradients from the closest four grid points are calculated. The resulting values are then interpolated to produce smooth noise. The more advanced Simplex Noise replaces the grid with a regular triangular grid, reducing the complexity in terms of dimensions and minimizing directional artifacts present in Perlin Noise based on typical gradients. The example of pure noise can be seen in Fig. 1. The structural differences are visually evident, and the potential advantage of this type of noise compared to standard Gaussian perturbation is intuitive: the distortions are more structured, and the denoising process will be able to "repair" those structural anomalies.

For the two-dimensional case, a coordinate transformation is first performed using the following formulas:

$$x' = x + (x + y) * F \text{ 錯誤! 尚未定義書籤。}, \quad (5)$$

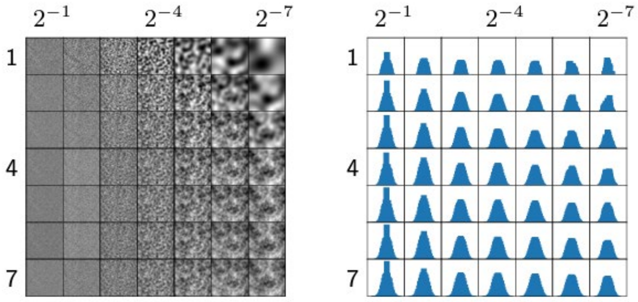
$$y' = y + (x + y) * F, \quad (6)$$

$$F = \frac{\sqrt{n+1}-1}{n}. \quad (7)$$

where *n* represents the dimension.

After the coordinate transformation, the Simplex lattice is determined, and the vertices of the Simplex are obtained. Then, gradient values are selected, and finally, a radial attenuation function is used to compute and sum the contribution values for each vertex. The function for calculating the vertex contribution value is given by:

$$\left(max(0,r^2-d^2)\right)^4 * \left(\langle\Delta x,\Delta y\rangle\cdot\langle\nabla x,\nabla y\rangle\right) \quad (8)$$



(a) Structures of simplex noise (b) Histograms of simplex noise

Fig. 1 Example of Simplex Noise.

C. GAN-based noise model

A GAN-based model is proposed for denoising real images, aiming to address the noise issues present in real-world images [3]. The model introduced a GAN noise generator, and its generator architecture is illustrated in Fig. 2. This generator takes the original image and a version of the image with added noise as inputs, allowing the GAN to learn the distribution of the noise. Specifically, the model estimates the exposure and readout noise parameters of the input image and computes the noise standard deviation for each pixel. This design enables the generator to produce more realistic denoising results, resulting in clearer and more detailed images.

To train this model, the research team first utilized a large synthetic noise dataset for pretraining, enabling the generator to learn various types of noise. Then, they fine-tuned the model using a small real noise dataset to better adapt to real-world noise characteristics.

In addition to the generator, the model also includes a discriminator. The discriminator takes the generated noise and real noise as inputs and computes the distribution difference between them. This process helps validate the authenticity and quality of the denoising results generated by the generator.

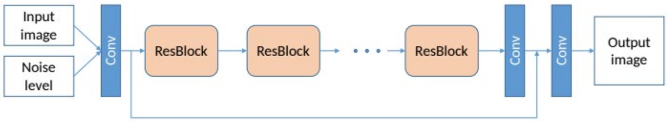


Fig. 2 GAN generator architecture.

$$Loss = E_{\tilde{x} \sim P_g}[D(\tilde{x})] - E_{\mathrm{x} \sim P_r}[D(\mathrm{x})] + \lambda E_{\hat{x} \sim P_{\hat{x}}}[(\|\nabla_{\hat{x}} D(\hat{x})\|_2 - 1)^2], \quad (9)$$

The loss function for GAN can be expressed in Eq.(9). Here, P_g represents the distribution of synthetic/generated data, and P_r represents the distribution of real data. The first term in the loss function calculates the expectation (E) of the discriminator's output for the generated noise, which corresponds to the discriminative result for the synthetic data. The second term calculates the expectation of the discriminator's output for real noise, representing the discriminative result for real data. The third term is a gradient penalty term, used to encourage the discriminator to maintain smooth gradients for real data, thereby improving training stability. λ is an adjustment parameter that balances the importance of different components.

By minimizing this loss function, the model aims to make the generated noise distribution closer to the distribution of real noise during the training process. This, in turn, helps improve denoising effects and the overall quality of the generated images.

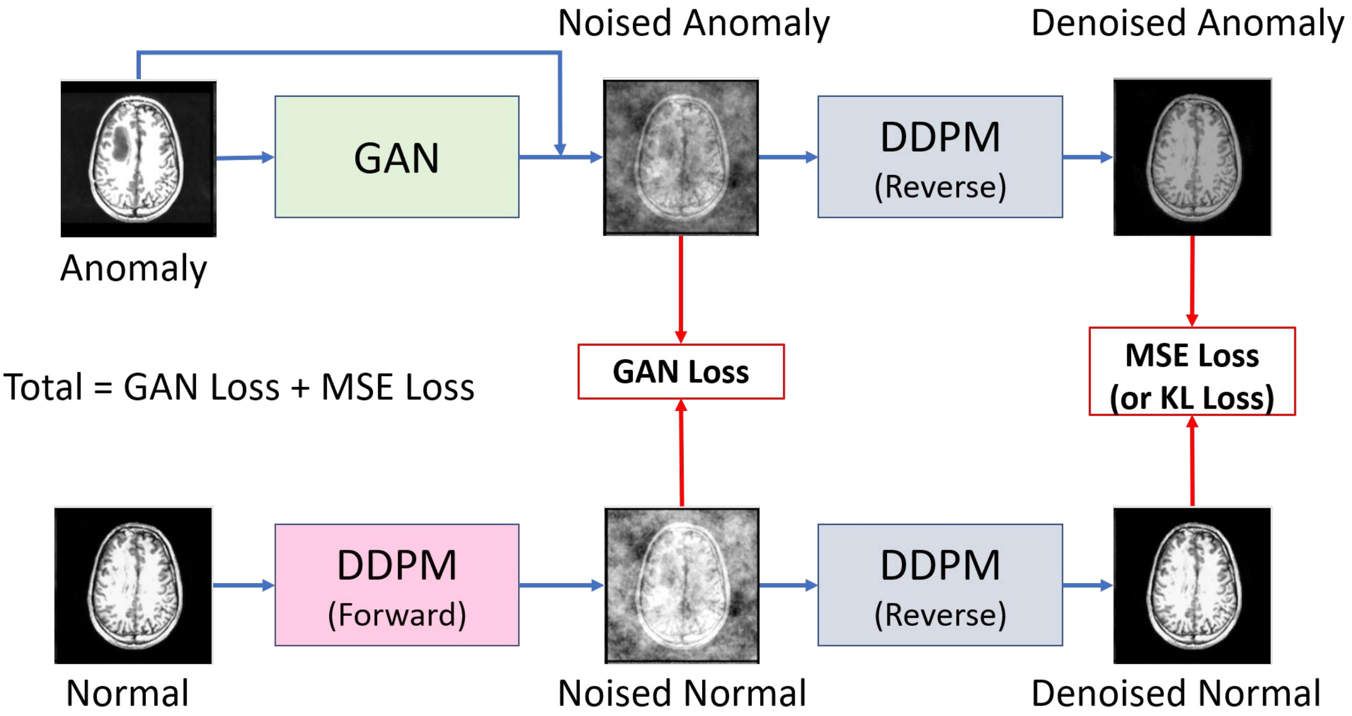


Fig. 3 Framework of SemanticDiffGAN.

III. PROPOSED METHOD

In this section, we describe in detail the framework of our proposed model.

A. *Overview*

The framework of our proposed model is shown in Fig. 3. The DDPM is pre-trained on normal data in order to take advantage of the noise generated by the forward and reverse processes of this DDPM to bootstrap the GAN model.

The anomalous images are turned into noisy images by generating corresponding noise from GAN and sent to the pre-trained DDPM model for denoising. The normal image is sent to the pre-trained DDPM to produce a noise added image and a noise removed image respectively.

B. *Loss function*

Our goal is to generate a special noise that can add noise to the input anomalous image to make it a noisy image that can be fed into the pre-trained DDPM model to generate results similar to normal images. For this purpose, we design two loss functions to achieve this goal.

GAN loss is the same as Eq. (9), and the purpose is to make the noise-added anomalous image as similar as possible to the normal image. MSE loss is the L2 similarity between the output result when a normal image is input and the output result when an abnormal image is input. The purpose is to guide the model to generate images as close to the normal result as possible.

In summary, the total loss of the proposed method is the GAN loss plus the MSE loss

IV. EXPERIMENT RESULTS

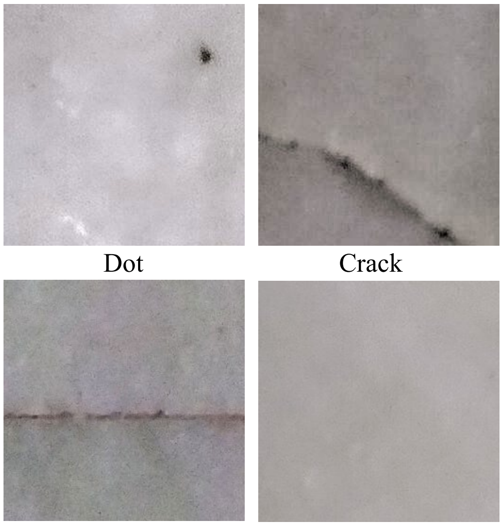
In this section we present the training results of our different model architectures, and analyze and discuss the results.

A. *Dataset and data pre-processing*

The dataset we used is called Marble Surface Anomaly Detection, which is a real-world dataset used for detecting anomalies in marble patterns. The dataset consists of two folders, namely train and test, each containing four types of images: Crack, Dot, Joint, and Good. The size of each image is 256 × 256. In the train dataset, there are a total of 2249 images, while in the test dataset, there are 688 images.

In this report, we only utilized images from three types: Crack, Dot, and Good, as the Joint category does not represent defective images. In the training set, there are 984 Crack images, 92 Dot images, and 860 Good images. Since we consider both Crack and Dot images as defective, we classify them as abnormal, while Good images are considered normal. In the original dataset, there are 1076 abnormal images and 860 normal images. However, in our model, we require one abnormal image and one normal image as inputs, so we need to augment the number of normal images to match the quantity of abnormal images.

To augment the number of normal images, we followed the same method as the author, which involves applying flips, rotations, and brightness adjustments to increase the diversity of the images. Since we need to add a total of 216 images, we randomly selected 216 images from the normal data and applied transformations to them. Specifically, we set the probability of horizontal flipping to 0.9, vertical flipping to 0.9, and performed random rotations of 90°, 180°, 270°, and 360°. Additionally, we randomly adjusted the brightness of the images to be 0.9 to 1.1 times the original image, creating new variations of the data.


Dot Crack

Joint Good

Fig. 4 Examples of Marble Surface Anomaly

B. *Pretrained DDPM*

T is set to 300, and β_t is set to evenly distribute between 0.0001 and 0.02 based on the setting of T. The noise follows a normal distribution N(0,1). The learning rate is set to 0.0003, optimizer is Adam, image size is 256, batch size is 8, and the training lasts for 100 epochs. To speed up the training process, during training, the value of T for each image is randomly chosen from 0 to 300. Only normal photos are used for training. The models are trained on NVIDIA GeForce GTX 1080 Ti. Fig. 5 is the result of our pretrained DDPM.

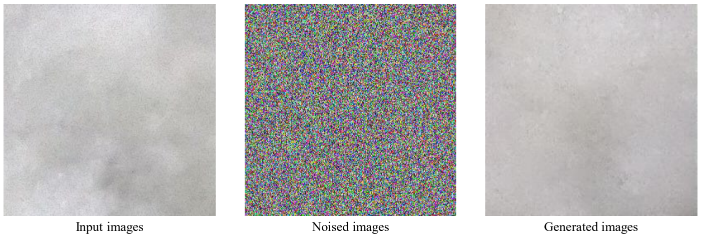

Input images Noised images Generated images

Fig. 5 Results of denoised anomaly images and noised anomaly images.

C. *Results of SemanticDiffGAN*

For this training, we set the parameters as follows: 5 epochs, a learning rate of 0.0005, and a batch size of 16. During training, the input images were resized to a dimension of 256×256. In this experiment, we successfully eliminated the defective parts and generated defect-free images (Denoised Anomaly). By subtracting the Denoised Anomaly from the original anomaly image, we obtained the Anomaly Map. Therefore, we can say that we have largely achieved our goal of finding a universal noise. Our model architecture can generate Noised Anomaly, Denoised Anomaly, Noised Normal, Denoised Normal. Fig. 6 and Fig. 7 are the results generated by our models.

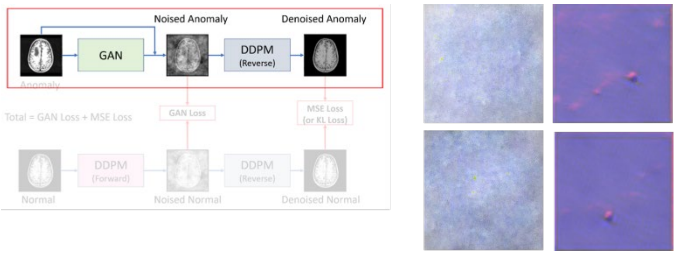


Fig. 6 Results of denoised anomaly images and noised anomaly images.

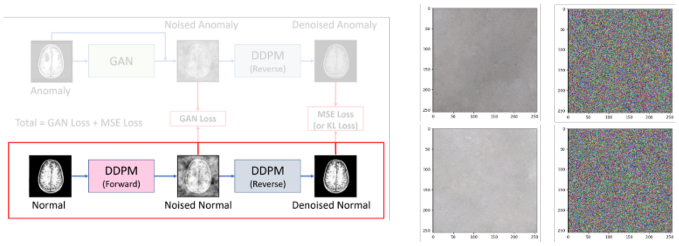


Fig. 7 Results of denoised normal images and noised normal images.

D. *Fail Cases*

Although we have successfully removed defects, our Denoised Anomaly and Noised Anomaly still struggle to learn certain cases well. Fig. 8 is one of our failure cases. We identified three possible reasons for this outcome. Firstly, our dataset consists of only around 1000 samples, which is relatively small. GANs may not have been able to effectively learn the necessary noise patterns. Secondly, we may have lacked a proper training strategy. As this was our first attempt at building and training a GAN, coupled with hardware limitations, we could only set the number of epochs to 5. Additionally, the learning dynamics between the generator and discriminator may not have been optimal. As suggested by the instructor, it may be necessary to incorporate multiple discriminators, but due to time constraints, we were unable to explore this aspect further. Thirdly, our generator model design may not have been optimal. During GAN training, we frequently encountered issues such as gradient explosion or vanishing gradients. To mitigate these issues, we speculate that adding an activation function constraint to the final output of the generator may be beneficial.

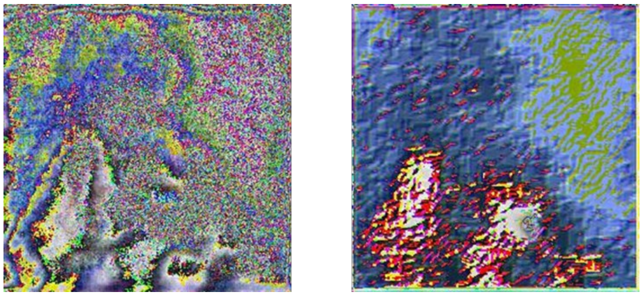


Fig. 8 Fail cases of denoised anomaly and noised anomaly.

V. CONCLUSION AND FUTURE WORK

Our proposed method was experimentally proven to be a feasible research direction, however, due to the time factor, we could not try different training strategies to achieve better results.

Due to the instability of the GAN model architecture training, a more appropriate learning strategy is needed. Another direction for improvement is to switch to a more stable noise estimation model or to improve the GAN model in the proposed method.

REFERENCES

[1] J. Ho, A. Jain, and P. Abbeel, "Denoising Diffusion Probabilistic Models," in *34th Conference on Neural Information Processing Systems (NeurIPS 2020)*, Vancouver, Canada.

[2] J. Wyatt, A. Leach, S. M. Schmon, and C. G. Willcocks, "AnoDDPM: Anomaly Detection With Denoising Diffusion Probabilistic Models Using Simplex Noise," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pp. 650-656, 2022.

[3] Tran, L.D., Nguyen, S.M., Arai, M. (2021). GAN-Based Noise Model for Denoising Real Images. In: Ishikawa, H., Liu, CL., Pajdla, T., Shi, J. (eds) *Computer Vision – ACCV 2020. ACCV 2020. Lecture Notes in Computer Science(), vol 12625*. Springer, Cham.

[4] OpenAI. "ChatGPT." OpenAI, 2021. https://openai.com.