Corrupted Fingerprint Recovering via Diffusion Model

Minrui Luo
IIIS, Tsinghua University
luomr22@mails.tsinghua.edu.cn

Fuhang Kuang
IIIS, Tsinghua University
kfh22@mails.tsinghua.edu.cn

Abstract

Fingerprints are important characteristics for recognizing a person. However, in many scenes we may encounter problems that the fingerprint is unclear, damaged, or incomplete. We introduces a method of recovering corrupted fingerprint images. We trained a diffusion model on large fingerprint dataset, then used it to recover the corrupted area of the fingerprint image. Finally, we compared how corrupted fingerprints and recovered fingerprints matches original fingerprints respectively via SourceAFIS fingerprint matcher[1]. We achieve notably increases in the matching score after recovering process.

1. Introduction

Fingerprint recognition is one of the most widely used biometric technologies for personal identification. It is based on the unique patterns of ridges and valleys on the surface of the fingertip. However, in real life, we may encounter problems that the fingerprint is unclear, damaged, or incomplete. In this case, the traditional fingerprint recognition method may not work well. To solve this problem, we trained a generative model that can reconstruct the fingerprint. Nowadays there are many fingerprint generative model, such as [1], [2], [3], and [4]. We leverage the power of diffusion model to complete the task.

2. Pipeline

The whole project consists of 4 parts.

2.1. Diffusion model training

We collected over ten thousand high-quality fingerprint images, and used them to train a diffusion model. We mainly use NIST special database[2] for model training, and FVC dataset [3] for testing.

[1]https://github.com/robertvazan/sourceafis-net
[2]https://www.nist.gov/itl/iad/image-group/nist-special-database-302
[3]http://bias.csr.unibo.it/fvc2000/

We use Diffusers[4] as our main codebase for training and inference. The model is a simple DDPM diffusion model trained to generate fingerprint images. The training process takes 10 hours on 4 RTX 3090s.

2.2. Add Noise and Recovering

We add Gaussian noise and stain to fingerprint images respectively.

We then apply the trained diffusion model to the corrupted fingerprint images. For images with Gaussian noise, we simply denoise them with last several hundred inference steps; for images with black stains, we first add Gaussian noise to the stained area, and then run inference steps. During the recovering, we only denoise the stained area, which makes result better.

2.3. Matching and Comparison

We select the open-source fingerprint matching library SourceAFIS to be the metrics of the effect of recovering. The algorithm takes 1:1 or 1:N images as input (here we use 1:1, that is comparing and matching two fingerprints), outputs their matching scores and decides whether they are the same fingerprint.

In order to quantitatively describe the effect of recovering, we proposed relatively matching score (RMS), which is the matching score (MS) between the new image and original image divided by the matching score between the original image and itself.

$$RMS = \frac{MS(img_{\text{target}}, img_{\text{origin}})}{MS(img_{\text{origin}}, img_{\text{origin}})} \tag{1}$$

This is more robust than the matching score since it has the ”self matching score” ad the benchmark, and images with different qualities varies in the value of ”self matching score”.

3. Result

We selected another dataset (FVC2000), added noise, then compared matching results of the noisy one and recov-

[4]https://github.com/huggingface/diffusers

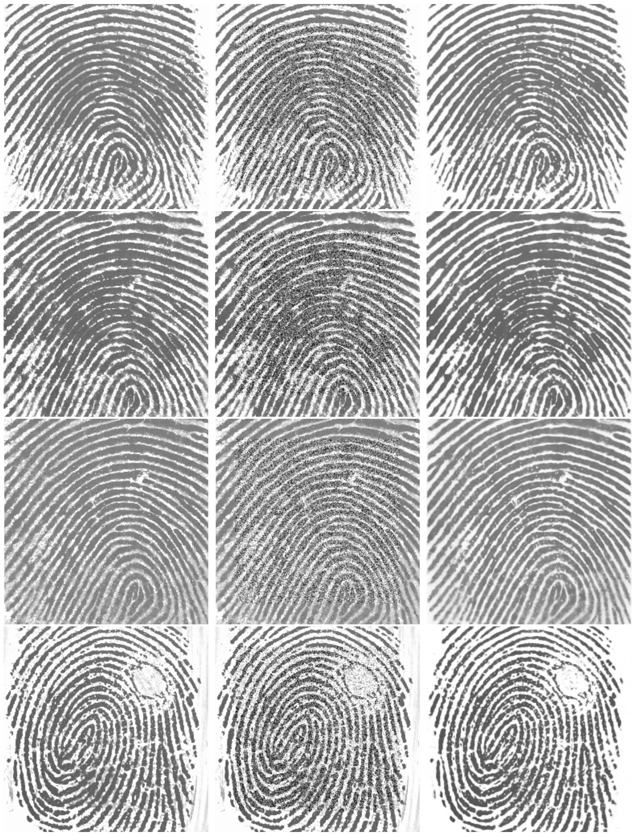

Figure 1. Recovering process, Gaussian noise. From left to right: original fingerprint $\rightarrow$ noisy $\rightarrow$ recovered

ered one from the original fingerprint respectively. We chose Gaussian noise and stain (a rectangular region of pure black) respectively.

3.1. Gaussian noise

In this section, we added each pixel by pixel-level-independent Gaussian noise. This process simulates application scenarios where sensors are damaged.

3.1.1 Recovering visualization

See Figure 1.

3.1.2 Matching scores

Histogram: see Figure 2

	Noisy RMS	Recover RMS
Mean	0.2462	0.4033
Variance	0.050	0.069

Improvement: 64%

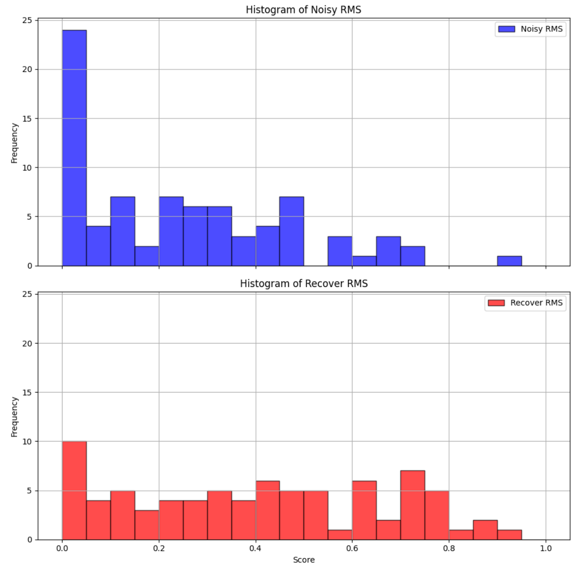

Figure 2. Histogram of Noisy RMS and Recover RMS when adding Gaussian noise

3.2. Stain

In this section, we filled a small rectangular region by pure black to simulate stain, that is, information of this region is lost completely. This process simulates application scenarios where fingerprints are fragmentary or stained.

To apply the method in the last section, we added a intermediate process of changing the pure black stain into pure pixel-level-independent Gaussian noise.

3.2.1 Recovering visualization

See Figure 3.

3.2.2 Matching scores

Histogram: see Figure 4

	Noisy RMS	Recover RMS
Mean	0.7297	0.7974
Variance	0.023	0.015

Improvement: 9.3% in mean RMS. We also note that the variance of RMS is lower. We assume it's because the stained area is random, and may evenly cover the more important area and less important area.

4. Conclusion and Future work

Conclusion. We use diffusion model to recover corrupted fingerprint images, and show that this can improve the matching accuracy. This shows the application prospect of

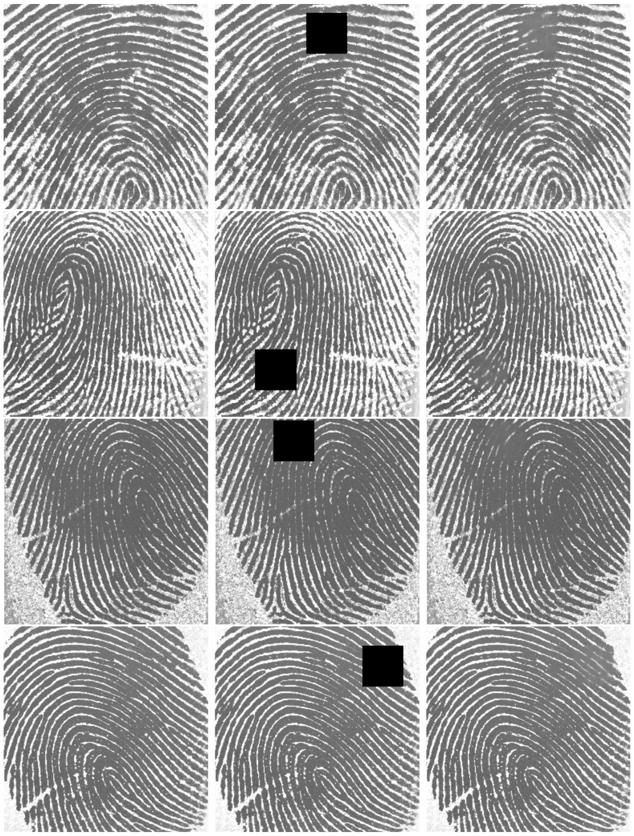

Figure 3. Recovering process, Stained. From left to right: original fingerprint → stained (→ replaced by Gaussian noise) → recovered

generative models such as diffusion model we used in the area of fingerprint recover and reconstruction.

However, we found that it cannot effectively recover images corrupted by Gaussian blurring (unless we add Gaussian noise on the area). We suppose that it is a principle problem of diffusion model: it only learns the Gaussian noise space, the images space, and the space between the two space. Maybe we can come up with some diffusion model that learns Gaussian blurring and de-blurring process, or we should try vision transformer.

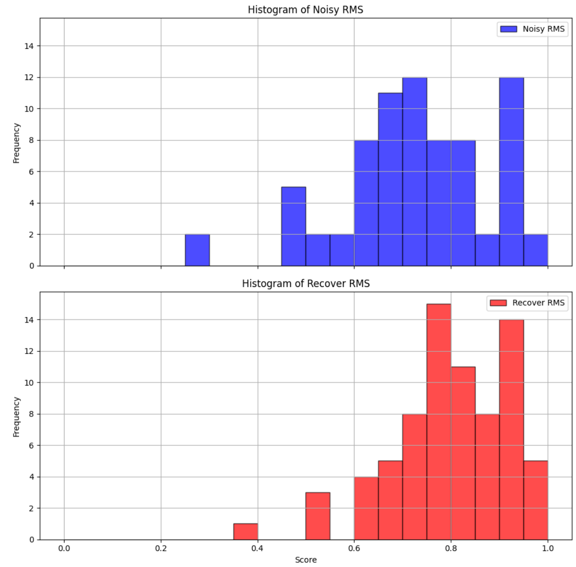

Figure 4. Histogram of Noisy RMS and Recover RMS when adding Stains

References

[1] Joshua J. Engelsma and Anil K. Jain. Generalizing fingerprint spoof detector: Learning a one-class classifier, 2019. 1

[2] Freddie Grabovski, Lior Yasur, Yaniv Hacmon, Lior Nisimov, and Stav Nimrod. Difffinger: Advancing synthetic fingerprint generation through denoising diffusion probabilistic models, 2024. 1

[3] Steven A. Grosz and Anil K. Jain. Universal fingerprint generation: Controllable diffusion model with multimodal conditions, 2024. 1

[4] W. Tang, D. Figueroa, D. Liu, K. Johnsson, and A. Sopasakis. Enhancing fingerprint image synthesis with gans, diffusion models, and style transfer techniques, 2024. 1