

Research on Deformed Character Generation Method based on CGAN

Tiantian Zhao, Linghua Guo*, Xiaofeng Zhao, and Fengshou Tao

College of Bioresources Chemical and Material Engineering, Shaanxi University of Science and Technology, Xi'an 710021, China

Abstract

A deformed character generation model based on Conditional Generative Adversarial Network (CGAN) was developed to reduce the risk of tampering with contract texts. Taking advantage of the characteristics of Chinese characters with "dot" strokes, which have high independence and good deformation concealment, a dataset of "dot" stroke deformed characters was constructed based on edge detection, providing training samples for the Conditional Generative Adversarial Network (CGAN). Based on this, a "dot" stroke deformed character generation model using CGAN was established to achieve the conversion from normal characters to "dot" stroke deformed characters. The model accuracy was verified using Structural Similarity Index (SSIM) and Peak Signal-to-Noise Ratio (PSNR). The "dot" stroke deformed characters were applied to anti-tampering of contract texts. The verification results show that compared with normal characters, the "dot" stroke deformed characters generated by the model have an average SSIM of 97.86% and an average PSNR of 19.72 dB, verifying the effectiveness of the model. Compared with manually designed "dot" stroke deformed characters, the average SSIM and PSNR are further improved to 98.44% and 28.58 dB, indicating that the model has high accuracy. After recognition of the replaced "dot" stroke deformed characters in contract texts, the average SSIM reaches 97.82% and the PSNR is 22.82 dB. The "dot" stroke deformed character generation model established in this paper has high accuracy, providing theoretical and practical references for the field of text anti-counterfeiting.

Keywords

Text Anti-counterfeiting; Deformed Characters; "Dot" Stroke Chinese Characters; Conditional Generative Adversarial Network (CGAN).

1. Introduction

With the rapid development of computer network technology and multimedia technology, the copying and circulation of information have become increasingly easy, leading to a growing risk of data text leakage. Documents, however, are one of the most efficient and direct methods for information recording, so preventing document leakage has always been a hot issue. In recent years, most efforts to solve document leakage have focused on the field of digital watermarking technology. With the rapid development of neural networks, adjusting geometric properties of characters (such as size, shape, or brightness) to carry watermark information is now intuitive and easy to understand^[1]. In the early stage, Liu Dong et al.^[2] first proposed the glyph topological structure watermarking technology. This technology realized the carrying of different watermark information by designing multiple variants of the same character and encoding them. However, this method has shortcomings, including poor visual effect, high cost of manual design, and low extraction and recognition rate. In recent years, deep learning-driven watermarking schemes have gradually emerged. Among them, Unet-GAN^[3] is

an image style transfer network based on Pix2Pix^[4]. It expands the number of convolutional layers to 16, which enables style conversion from printed text to handwritten text while maintaining the structural details of characters. Xiao et al.^[5] designed "FontCode", which embeds watermarks by applying slight disturbance to the outlines of English characters. This method is suitable for letters and numbers but difficult to handle Chinese characters with complex topology. To address the problem of generating Chinese deformed characters, Sun Shan^[6] proposed a Chinese document watermarking scheme that can automatically generate a Chinese deformed character library. This scheme adopts the currently popular Generative Adversarial Network (GAN) idea to generate Chinese character libraries. A noise layer was introduced into the network to enhance robustness in printing and scanning environments. However, this method damages the integrity of the original layout when modifying document content, fails to accurately locate the embedded deformed characters, and its recognition efficiency is also difficult to guarantee. Therefore, this study proposes a "dot" stroke Chinese deformed character generation model based on Conditional Generative Adversarial Network (CGAN). This model only deforms the "dot" strokes of Chinese characters, which reduces the difficulty of generation and makes it suitable for the field of text anti-counterfeiting. In addition, the model can solve the problem of complexity in Chinese character design and generation, and is finally applied to prevent tampering in contract texts.

2. Deformed Character Generation Model based on Conditional Generative Adversarial Network

2.1 Basic Principle

2.1.1 Basic Principles of Generative Adversarial Network (GAN)

A Generative Adversarial Network (GAN) consists of a generative model (Generator) and a discriminative model (Discriminator)^[7], as shown in Figure 1. Among them, the task of the discriminative model is to determine whether a given image is real and natural, that is, whether it is artificially generated. By contrast, the task of the generative model is to generate real and natural images, and to make these generated images as consistent as possible with the distribution of original data. During the training process, the generative model and the discriminative model form a dynamic game process. Both models are continuously optimized in the iteration process. When the discriminative model can no longer distinguish between real data and generated data, the generative model can be considered to have reached an approximately optimal state.

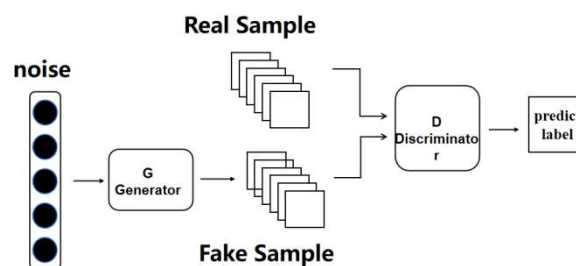


Figure 1. Architecture of Generative Adversarial Network (GAN)

2.1.2 Basic Principles of Conditional Generative Adversarial Network (CGAN)

The Conditional Generative Adversarial Network (CGAN) is a generative model that incorporates additional conditional information y into both the generator and the discriminator. By doing so, the model is able to learn the conditional data distribution $p(x|y)$ and generate samples x that match the given condition y . The core idea is to conduct adversarial training, where the generator produces realistic samples under the constraint of y , while the discriminator, also guided by y , distinguishes between real samples and those generated by the model^[8].

The objective function of CGAN is an adversarial process, and its formula is shown in Equation (1):

$$\min_G \max_D V(D, G) = E_{(x,y) \sim p_{data}(x,y)} [\log D(x, y)] + E_{z \sim p_z(z), y} [\log(1 - D(G(z, y), y))] \quad (1)$$

In the formulation, x represents the real sample data, y denotes the conditional information, and z is the input noise to the generator. The generator $G(z, y)$ produces samples by combining the noise z with the condition y , while the discriminator $D(x, y)$ evaluates the probability that a given sample x is real (output close to 1) or generated (output close to 0) under the same condition.

In this study, as shown in Figure 2, the manually designed deformed characters with the “dot” stroke are used as conditional input to guide the training process. The model is designed to map standard character images to their corresponding deformed versions. The training process combines adversarial learning with pixel-level loss: the generator G learns to produce deformed characters from standard inputs that resemble the manually designed targets y , while the discriminator D distinguishes between the generated outputs $G(x)$ and the real conditional samples y .

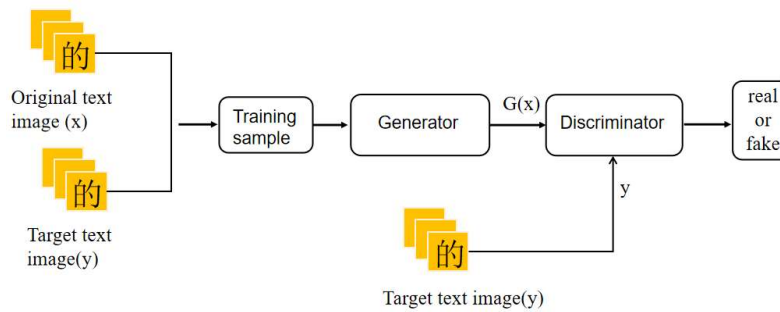


Figure 2. Conditional Generative Adversarial Network

2.2 Establishment of Dataset based on Conditional Generative Adversarial Network (CGAN)

2.2.1 Establishment of Normal Text Dataset Based on Edge Detection Method

Image edge detection is a key technology in computer vision and image processing for identifying the contours of objects or regions^[9]. According to the "GB 13000.1 Standard for Chinese Character Order (Stroke Order) in Character Set"^[10], Chinese character strokes can be divided into five categories: horizontal, vertical, left-falling, dot, and folded strokes. Considering the feasibility and concealment of structural deformation, this study focuses on the deformation design of the "dot" strokes in Chinese characters. The "dot" stroke has the advantages of high independence, concealed deformation, and little impact on the overall character shape. It can not only achieve slight visual deformation while maintaining readability, but also facilitates machine extraction of deformation features. Therefore, it has higher practical value and research significance in the design of deformed characters.

To design Chinese characters with "dot" strokes into deformed characters for sample set construction, firstly, it is necessary to locate the characters containing "dot" strokes, and then identify the optimal combination of Chinese characters with "dot" strokes that is suitable for the CGAN model from all such characters. The specific steps for dataset construction using the edge detection method are as follows:

1) Preprocessing:

A total of 3,500 commonly used Chinese characters were selected. Each image was first converted to grayscale and processed with Gaussian filtering to reduce noise. Then, Otsu's global adaptive thresholding was applied, and the grayscale values of the characters and background were inverted, ensuring that the foreground was white and the background was black.

2) Contour detection:

On the preprocessed images, the outermost contours were extracted (as shown in Figure. 3). Each connected component was traversed, and its pixel area was calculated.

3) Defining the stroke area of the “dot”:

Based on a statistical analysis of individual stroke areas, the pixel range for the “dot” stroke was determined to be between 530 and 1100 pixels.

4) Screening characters with the “dot” stroke:

Using this area range, morphological opening operations were performed to separate the “dot” from other strokes. From the 3,500 characters, a total of 1,129 characters containing the “dot” stroke were accurately identified (detection results shown in Figure. 4).

5) Selecting the optimal dataset for model training:

Taking contract documents as the research object, word frequency statistics were performed on their text content. The 1,648 most frequent characters were extracted as high-frequency candidates. The intersection of this set with the previously screened “dot” stroke characters yielded 524 characters. These characters, which both contain the “dot” stroke and frequently occur in contracts, were chosen as the optimal candidates for substitution in later experiments.



Original image, Gaussian fuzzy desaturation, Gray inversion, Contour detection

Figure 3. Text contour detection



Figure 4. Locate the "dot" strokes of the text

2.2.2 Establishment of the "Dot" Stroke Deformed Character Dataset Based on the Edge Detection Method

For the 524 high-frequency characters containing the “dot” stroke, the design process is illustrated using the character “的” as an example:

1) Locating the “dot” stroke:

Each of the 524 character images was first loaded and converted from the RGB color space to HSV. A stroke mask was then generated, as shown in Figure 5, to accurately identify the “dot” strokes highlighted in red.

2) Transforming the “dot” stroke:

Using the mask, the red “dot” stroke was separated from the original image, and its original position was filled with a white background. The “dot” stroke was treated as an independent patch and randomly shifted by ± 10 pixels along both the X and Y directions, resulting in the deformed samples shown in Figure 5.

3) Encoding the deformed “dot” strokes:

A binary coding scheme was applied to map each of the four displacement directions to watermark information: “00” for upward, “01” for downward, “10” for leftward, and “11” for rightward

movement. For example, applying these transformations to the “dot” stroke of the character “的” produced four distinct deformed versions. These deformations preserved the overall readability and aesthetics of the character while ensuring reliable recognition and extraction when substituted into contract documents.

Based on this process, a training dataset was constructed, consisting of 524 standard character images and 4×524 deformed character images with modified “dot” strokes. The standard characters were grouped as set A, and the manually designed deformed characters as set B.

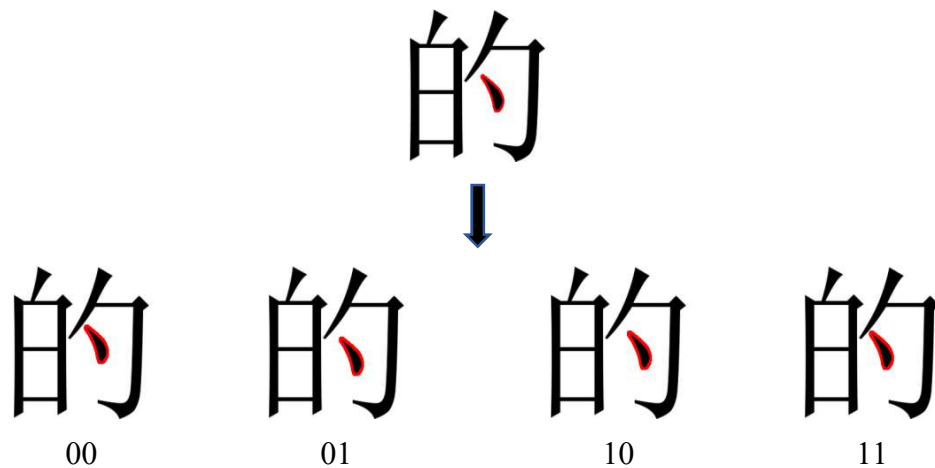


Figure 5. Design deformed text

2.2.3 Reliability of the Dataset

The deformed character dataset with modified “dot” strokes was constructed using the edge detection method. This approach relied on predefined rules, such as edge thresholds and fixed deformation operations, to generate samples in a deterministic manner. In terms of quantity, previous studies have shown that around 400 samples are sufficient for supervised image-to-image translation tasks, which validates the feasibility of this dataset size. In this work, 524 samples were collected for the “dot” stroke deformation task, ensuring reliability at the data level. However, this dataset was used only during the data preparation stage to provide the Conditional Generative Adversarial Network (CGAN) with high-quality, standardized, and large-scale training samples. The CGAN, as an intelligent generative model, then learns the mapping from standard Chinese characters to deformed characters with modified “dot” strokes, achieving end-to-end and scalable generation. Unlike the deterministic dataset construction, the CGAN performs probabilistic generation by learning data patterns and optimizing output quality. Therefore, the dataset construction and the CGAN-based generation correspond to different stages of the workflow-data preparation and model application-without any methodological conflict.

2.3 Determination of the Optimal Environment Based on Conditional Generative Adversarial Network (CGAN)

2.3.1 Determination of the Model

The generator was implemented using a U-Net encoder–decoder architecture. The discriminator was designed based on the PatchGAN framework, which combines convolutional hierarchies with input modules, and was further optimized for this study. The original PatchGAN discriminator consists of five convolutional blocks, with the number of feature channels (ndf) starting at 64 and doubling at each layer. However, to prevent the discriminator from becoming overly powerful-leading to faster convergence and suppressing the generator, which would cause instability in adversarial training-we applied a lightweight modification. Specifically, the number of convolutional layers was reduced to four, including two down-sampling layers and two output layers, and the initial channel number ndf was decreased from 64 to 32. The resulting lightweight discriminator (shown in Fig. 6) was

experimentally validated to effectively distinguish between real and generated images, while significantly improving training efficiency without noticeable performance degradation. For input, the normal character image x was concatenated with the corresponding deformed “dot” stroke image—either the real sample y or the generated output $G(x)$ —ensuring that the discriminator could simultaneously access the conditional information.

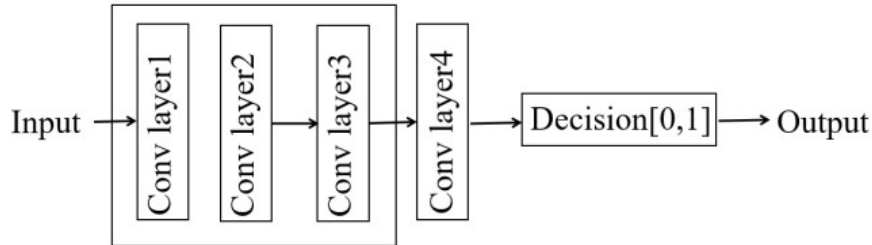


Figure 6. Network structure of lightweight discriminator

2.3.2 Configuration of Loss Function

The loss function of the Conditional Generative Adversarial Network (CGAN) consists of two parts: the generator loss and the L1 loss. The generator loss corresponds to the adversarial loss used in the original GAN framework. In the process of transforming a normal character image x into its manually designed deformed version with the “dot” stroke y , the generator G produces a synthetic image $G(x)$. The discriminator D then evaluates whether the input image comes from the real deformed sample y or the generated output $G(x)$. This establishes the adversarial process, where the generator loss encourages G to produce images that are indistinguishable from the real samples, making it difficult for D to differentiate between $G(x)$ and y . The formulation of the generator loss is expressed as follows:

$$L_{cGAN} = E_{x,y} [\log D(x, y)] + E_x [\log(1 - D(x, G(x)))] \quad (2)$$

In this context, x and y denote the input normal text image and the manually designed “dot” stroke variant image, respectively. $G(x)$ represents the generated synthetic image, where G is the generator and D is the discriminator.

When relying solely on the generator loss, the generated images often contain rich texture details but may fail to maintain global consistency or appear blurred. To address this issue, a pixel-level L1 distance penalty is introduced, which measures the mean absolute error between the generated image and the ground truth in the pixel space. The L1 loss function is expressed as follows:

$$L_{L1}(G) = E_{x,y} [\|y - G(x)\|_1] \quad (3)$$

Therefore, the overall loss function of this model is as follows:

$$L(G, D) = L_{cGAN}(G, D) + \lambda L_{L1}(G) \quad (4)$$

In the formula: λ is the weight coefficient, which is used to balance the relative importance of the two losses.

2.3.3 Training Parameters and Hardware Environment

In this study, the Generative Adversarial Network was implemented using the PyTorch framework, with the Adam optimizer employed for model training. The initial learning rate was set to 0.0001, which falls within the commonly effective range for CGAN-based image translation tasks (10^{-5} – 10^{-3}). The learning rate was reduced by half every 50 epochs, and the training was conducted for a total of 100 epochs. During the early stage (epochs 0–50), the initial learning rate was applied for coarse adjustment, while in the later stage (epochs 51–100), a reduced learning rate of 0.00005 was used for fine-tuning. This strategy ensures stable model convergence while preventing overfitting caused by excessive training on a relatively small dataset. The experiments were carried out on the Ubuntu 22.04.3 LTS operating system, using an Intel (R) Xeon (R) Platinum 8269CY CPU @ 2.50GHz, with GPU acceleration provided by four NVIDIA GeForce RTX 3060 cards.

2.4 Evaluation Metrics

In the experiments, the normal text images from the training set were used as the test set, while the manually designed “dot” stroke variant images served as the target set. These data were fed into the trained CGAN model, and the variations of the generator adversarial loss (G_GAN) and the L1 loss during iterations were recorded. The test samples were then re-input into the trained model, and both subjective evaluation and objective evaluation were applied to analyze the results. This process was conducted to verify the effectiveness of the generated images as well as the accuracy of the model.

2.4.1 Subjective Evaluation

To assess the consistency between the “dot” stroke variant characters generated by the model and those manually designed from the perspective of human visual perception, a subjective evaluation experiment was conducted. A total of 20 graduate students with normal (or corrected-to-normal) vision were recruited as evaluators, including 8 males and 12 females, aged between 22 and 26. All participants had prior knowledge of Chinese character structures but no previous experience with similar experiments.

Ten representative Chinese characters (such as “的”, “付”, “记”, “辽”) were selected from the training set as evaluation samples. Each sample group included three types of images: the original normal character, the manually designed “dot” stroke variant, and the model-generated “dot” stroke variant. All images were standardized to a resolution of 256×256 pixels and presented in grayscale on a display device with 1920×1080 resolution. The lighting in the experiment environment was controlled at 500±50 lux to minimize reflection interference.

The evaluation framework was based on four core dimensions, using a five-point Likert scale (where 1 indicated “significant difference,” 3 indicated “minor difference,” and 5 indicated “complete consistency”). The evaluation dimensions were as follows:

- 1) Overall visual similarity: assessing the overall visual match between generated and manually designed characters.
- 2) Accuracy of “dot” stroke variation: focusing on the positional shift, shape integrity, and consistency of the “dot” stroke with the manually designed reference.
- 3) Structural integrity of the character: determining whether the generated character maintains the original skeleton and recognizability.
- 4) Visual naturalness: evaluating whether the stroke transitions and overall harmony of the generated character align with natural handwriting patterns.

2.4.2 Objective Evaluation

In this experiment, the optimal result of the generated “dot” stroke variant characters is expected to be nearly identical to the manually designed counterparts. To evaluate the differences between the generated images and the manually created variants, Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index (SSIM) were adopted as the objective evaluation metrics.

- 1) PSNR

PSNR is an objective criterion commonly used for assessing image quality, which is derived by applying a logarithmic calculation based on the mean squared error (MSE). In this study, PSNR is employed to evaluate the quality of the generated “dot” stroke variant character images. The calculation formula for the mean squared error is given as follows:

$$MSE = \frac{1}{nm} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [I(i, j) - K(i, j)]^2 \quad (5)$$

Among them, I and K represent two images with a size of $m \times n$. PSNR can be calculated using Equation (6):

$$PSNR = 10 \log_{10} \left(\frac{MAX_i^2}{MSE} \right) \quad (6)$$

The larger the PSNR value calculated by the formula is, the better the quality of the generated "dot" stroke deformed character images is, and its value range is between 0 and 100.

2) SSIM

Structural Similarity Index (SSIM) compares the differences between generated "dot" stroke deformed character images and manually designed "dot" stroke deformed character images from three aspects: brightness, contrast, and structure. During the calculation process, for each group of images, the mean value of the image is subtracted to reduce the impact of brightness on the results; meanwhile, the normalized variance of the image is used to avoid the impact of contrast. Finally, the structural information of the image is obtained. The value of SSIM is defined to range between 0 and 1, and a larger value indicates that the two images are more similar in structure.

Formula for brightness calculation:

$$l(x, y) = \frac{2\mu_x\mu_y + c_1}{\mu_x^2 + \mu_y^2 + c_1} \quad (7)$$

Formula for Contrast Calculation:

$$c(x, y) = \frac{2\sigma_x\sigma_y + c_2}{\sigma_x^2 + \sigma_y^2 + c_2} \quad (8)$$

Formula for Structure Calculation:

$$s(x, y) = \frac{\sigma_{xy} + c_3}{\sigma_x\sigma_y + c_3} \quad (9)$$

Formula for Structural Similarity Calculation:

$$SSIM(x, y) = [l(x, y)^{\alpha} c(x, y)^{\beta} s(x, y)^{\gamma}] \tag{10}$$

3. EXPERIMENTS

3.1 Experimental Process

1)Data Preprocessing:

PIL.Image.open is used to load images and convert them to RGB format. A single image is split into two parts from the middle: Part A (normal characters) and Part B (manually designed "dot" stroke deformed characters). The same transformations are applied to each part respectively to ensure pairing consistency.

2)Model Training and Optimization Process:

The generator network is responsible for converting input normal character images into "dot" stroke deformed character images, while the discriminator network evaluates the consistency between the generated results and the manually designed "dot" stroke character data. To suppress overfitting, multiple loss functions are introduced during the training process, including L1 regularization loss, generative adversarial loss, and discriminative loss. These loss functions jointly optimize the parameters of the generator network to improve the authenticity and style consistency of the generated images. At the same time, the discriminator network calculates the loss based on the judgment results of the generated samples, and optimizes its own parameters through backpropagation. This enhances its ability to distinguish between manually designed "dot" stroke deformed character images and generated "dot" stroke deformed character images. Finally, through the adversarial learning and collaborative optimization of the generator and discriminator networks, the synthesis of high-quality "dot" stroke deformed characters and their effective discrimination are achieved.

3.2 Experimental Results

The experiment takes the normal text image dataset A and the manually designed "dot" stroke deformed character image dataset B as inputs, and simultaneously records the generator's adversarial loss (G_GAN Loss) and L1 loss (G_L1 Loss). As shown in Figure 7, as the number of iterations increases, G_GAN Loss decreases continuously and finally drops below 1, which indicates that the authenticity of the images generated by the generator continues to improve. The L1 loss in Figure 8 also undergoes a process from rapid decline to slow convergence, and stabilizes at approximately 0.3–0.4 when approaching 100 iterations. This shows that the generation effect of pixel-level details is significantly improved.

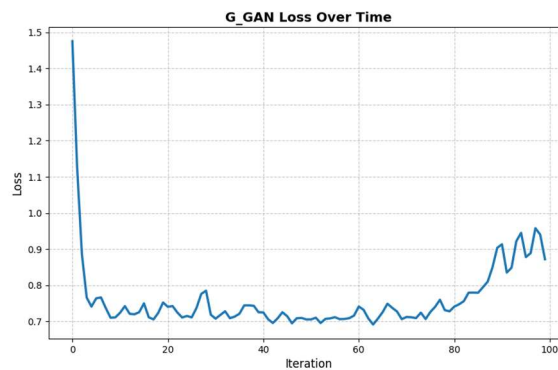


Figure 7. Generator loss

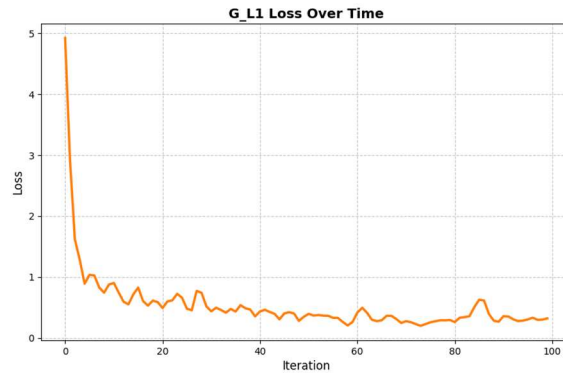


Figure 8. L1 loss

The generation effect of converting normal text images to "dot" stroke deformed character images is shown in Figure 9. Four Chinese characters are randomly selected for the display of generation results:

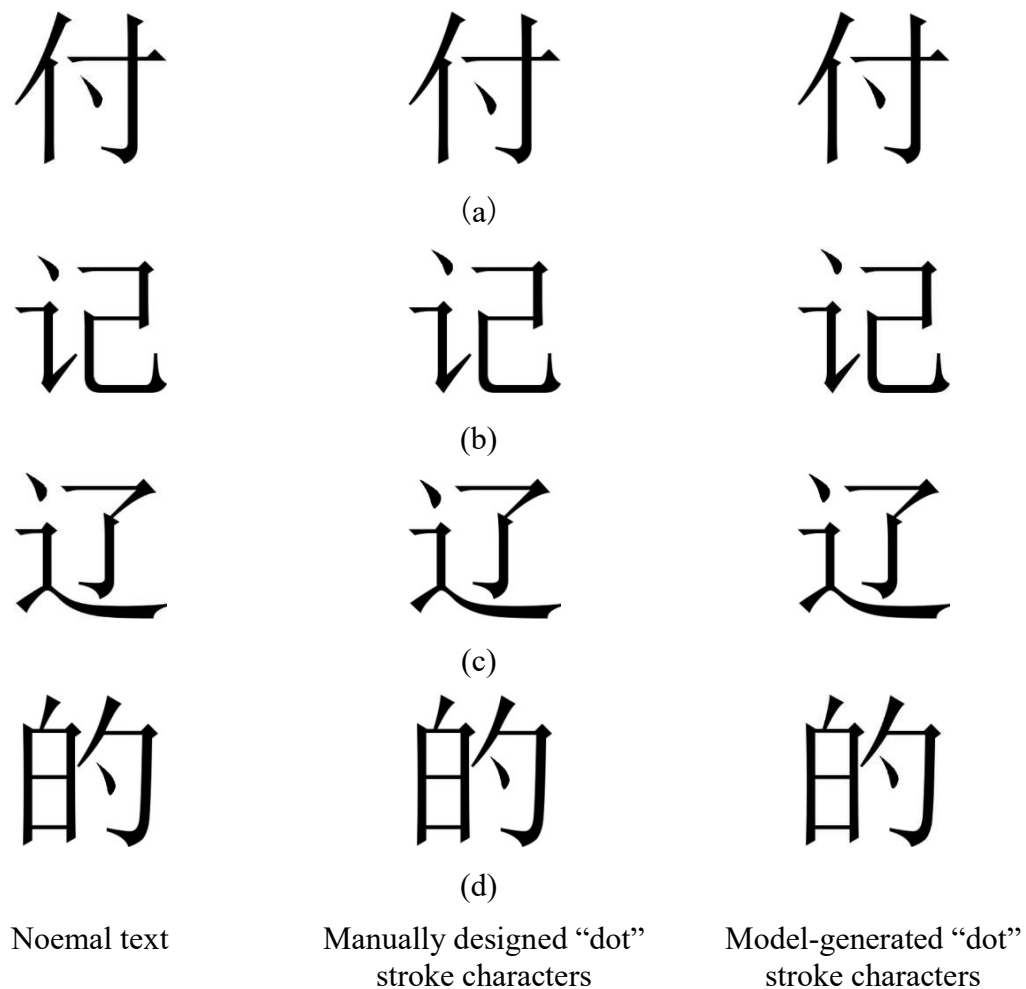


Figure 9. Display of style migration results

3.3 Evaluation of Experimental Results

The experimental results were quantitatively evaluated using subjective evaluation and objective evaluation. First, the "dot" stroke deformed characters automatically generated by the trained model were compared with normal characters to verify the effectiveness of deformation. Then, they were compared with manually designed "dot" stroke deformed characters to verify the generation accuracy of the model.

In the subjective evaluation process, samples were presented using a random blind method. Each evaluator completed the scoring of all samples independently, and the interval between repeated scorings for the same sample was no less than 24 hours to reduce memory bias. Finally, the mean score and standard deviation of the scores from 20 evaluators were taken, and the statistics are summarized in Table 1 below:

Table 1. Results of Subjective Evaluation

Evaluation Indicators	Overall Visual Similarity	Deformation Accuracy of "Dot" Strokes	Integrity of Character Structure	Visual Naturalness	Comprehensive Evaluation
Score	4.72±0.31	4.68±0.35	4.81±0.27	4.70±0.30	4.70±0.30

Additionally, the "dot" stroke deformed characters automatically generated by the trained model were compared with normal characters to further verify the effectiveness of deformation. The specific values of the objective evaluation are shown in Table 2.

Table 2. Objective evaluation details

Image	SSIM	PSNR
a	0.969548741	20.05269242
b	0.972621547	20.25241901
c	0.971587333	20.00848305
d	0.971398423	20.69082126
Average	0.971289011	20.251103935

The evaluation metrics of four sample images are presented in Table 2. In this experiment, the SSIM and PSNR values were calculated for fifteen images, and their average values were obtained. The summarized results are shown in Table 3.

Table 3. Objective evaluation details of 15 images

Image	SSIM	PSNR
1	0.967171152	18.88144815
2	0.967791968	20.1727509
3	0.972991563	19.91574889
4	0.968964252	20.01514978
5	0.969658415	19.54405032
6	0.968834776	19.34899103
7	0.969548741	20.05269242
8	0.969327443	19.31786129
9	0.972439118	20.15987266
10	0.97117238	20.15378828
11	0.972621547	20.25241901
12	0.971587333	20.00848305
13	0.94640041	17.25946427
14	0.968364833	19.99049308
15	0.971398423	20.69082126
Average	0.96855149	19.71760229

To verify the generation accuracy of the model, the "dot" stroke deformed characters automatically generated by the trained model were compared with the manually designed "dot" stroke deformed characters. The specific values of the objective evaluation are shown in Table 4.

Table 4. Objective evaluation details

Image	SSIM	PSNR
a	0.989284	29.909377
b	0.989762	29.854922
c	0.989957	29.765386
d	0.989562	30.49354
Average	0.98964125	30.00580625

Table 4 above presents the evaluation indicators for four sample images. In this experiment, the Structural Similarity Index (SSIM) and Peak Signal-to-Noise Ratio (PSNR) were calculated for fifteen images respectively, and their average values were obtained. The results are summarized in Table 5.

Table 5. Objective evaluation details of 15 images

Image	SSIM	PSNR
1	0.9844	28.582248
2	0.988368	30.058348
3	0.990746	29.188561
4	0.988683	29.863249
5	0.990424	30.126203
6	0.989284	29.909377
7	0.988797	28.782291
8	0.990676	29.982608
9	0.989025	29.896962
10	0.989762	29.854922
11	0.989957	29.765386
12	0.983117	27.224001
13	0.987790	29.672767
14	0.989562	30.49354
15	0.988962	29.518306
Average	0.9844	28.582248

Experimental results show that the average Structural Similarity Index (SSIM) and Peak Signal-to-Noise Ratio (PSNR) between the "dot" stroke deformed characters generated by the model and normal characters reach 97.86% and 19.72 respectively, verifying the effectiveness of the model. Compared with manually designed "dot" stroke deformed characters, the average SSIM and PSNR are further

increased to 98.44% and 28.58, indicating that the model has high accuracy, further verifying that the images generated by this model have high quality and accuracy.

The deformation effects of the "dot" stroke deformed characters automatically generated by the trained model in four directions (up, down, left, and right) are shown in Figure 10. Four Chinese characters were randomly selected from the test set to display the generation results.



Upward shife Downward shife Shife left Rightward shift

Figure 10. Display of style migration results

4. Anti-Counterfeiting Verification of "Dot" Stroke Deformed Characters in Contract Texts

4.1 Localization of "Dot" Stroke Deformed Characters

To replace the encoded "dot" stroke deformed characters automatically generated by the model with normal characters in contracts for anti-counterfeiting and traceability, it is necessary to determine the positions of normal characters to be replaced in contract documents in advance. Therefore, we need to localize the positions for replacing deformed characters in contracts.

This study takes contract documents in PDF format as the processing object, and the specific experimental steps are as follows:

1) Contract Document Format Conversion and Preprocessing:

Taking contract documents in PDF format (with font set to SimSun, Size 4) as the processing object, convert them into binarized JPG images. This provides basic image data for subsequent text line localization and segmentation.

2) Text Line Localization Based on Horizontal and Vertical Projection Method:

For each converted binarized image, the horizontal and vertical projection method is adopted to traverse and count the pixels in each line of the image (the statistical results are shown in Figure 11). By analyzing the pixel distribution characteristics, the start and end positions of each text line are accurately localized.

3) Text Line Segmentation and Horizontal Reference Line Drawing:

Based on the text line boundary information localized in Step 2, horizontal reference lines are drawn on the image to complete the segmentation of contract text lines (the segmentation effect is shown in Figure 12).

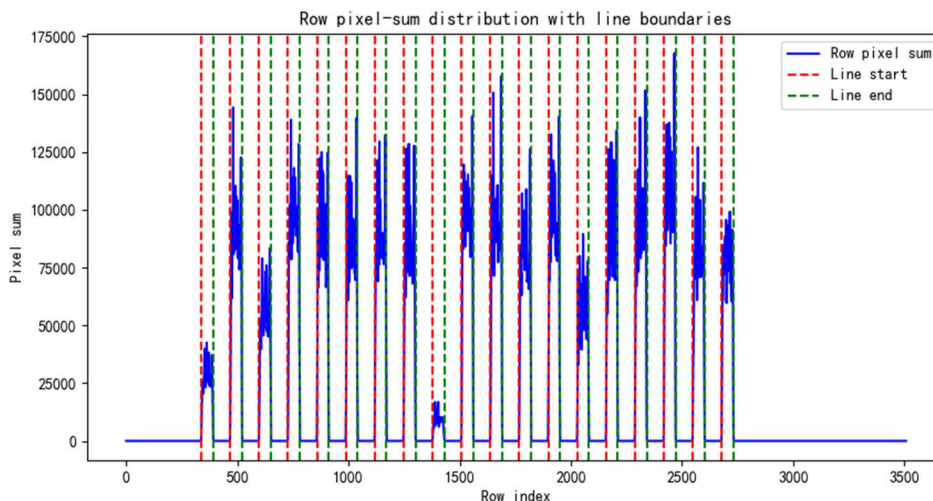


Figure 11. Text line pixels and variation map

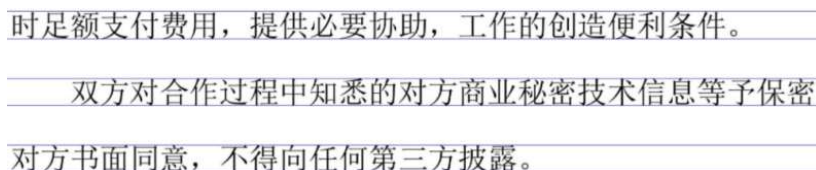


Figure 12. Detail image of text line segmentation

4) Preliminary Extraction of Character Columns in Text Lines:

Traverse all pixels in the segmented text lines to conduct preliminary extraction of character columns. Combined with the character size characteristics (under 300 DPI, the height of SimSun Size 4 Chinese characters is approximately 50–60 px, and the character spacing is approximately 1.333 px), to avoid splitting the same character, the actual thresholds are set as follows: the Chinese character height threshold is 54 px, and the character spacing threshold is 1 px.

5) Determination of Character Horizontal Intervals:

Starting from each non-zero pixel in the text line, perform iterative scanning along the horizontal direction with a step size of 1 px and a scanning length of 54 px until reaching the end of the image. Through this process, the horizontal interval of each character is determined.

6) Drawing of Character Bounding Boxes:

According to the character horizontal intervals determined in Step 5, draw corresponding bounding boxes on the original image, thus completing the accurate localization of characters (the results are shown in Figure 13).

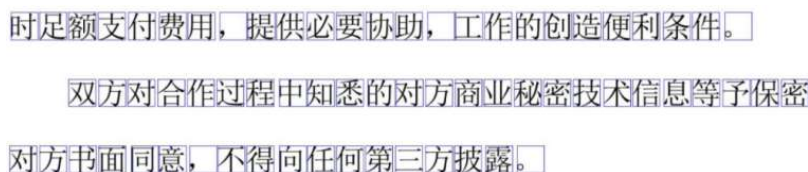


Figure 13. Detailed image of text column segmentation

Text Recognition in Bounding Boxes and Information Association: Use Optical Character Recognition (OCR) to recognize the text within each bounding box, and associate the positional information of the bounding boxes with the text inside them. The bounding box images and positional information of individual characters are output as shown in Table 6. Thus, all positional information of the text in the contract can be accurately obtained.

Table 6. Characters and their positional information

Character	Top Left	Top Right	Bottom Left	Bottom Right
技	(1009,338)	(1062,338)	(1009,391)	(1062,391)
术	(1068,338)	(1121,338)	(1068,391)	(1121,391)
服	(1125,338)	(1178,338)	(1125,391)	(1178,391)
务	(1184,338)	(1237,338)	(1184,391)	(1237,391)
合	(1242,338)	(1295,338)	(1242,391)	(1295,391)
作	(1300,338)	(1353,338)	(1300,391)	(1353,391)
协	(1359,338)	(1412,338)	(1359,391)	(1412,391)
议	(1418,338)	(1471,338)	(1418,391)	(1471,391)
鉴	(494,468)	(547,468)	(494,521)	(547,521)
...

In this study, four high-frequency characters that also exist in the character library were selected from the contract: "约 (Yue)" (appearing 14 times), "协 (Xie)" (appearing 11 times), "的 (De)" (appearing 9 times), and "付 (Fu)" (appearing 7 times). Their positions in the document were accurately located based on the previously extracted position coordinates, so as to determine the target areas for embedding deformed characters.

4.2 Replacement of "Dot" Stroke Deformed Characters

1) Construct a "Character-Watermark-Deformed Character" Mapping Table:

To realize the automatic replacement of deformed characters, this study first constructed a "Character → Watermark → Deformed Character" mapping table. It establishes a one-to-one correspondence between Chinese characters, binary watermark information, and corresponding deformed character images, providing an index basis for the subsequent replacement process.

2) Binary Code Segmentation and Deformed Character Image Matching & Loading:

For the input 8-bit binary code (e.g., 01100011), it is split into 4 subcodes. Each subcode corresponds to one "dot" stroke deformation direction (upward translation, downward translation, leftward translation, rightward translation). Based on the subcode information, the corresponding deformed character images are matched and loaded in sequence through the mapping table constructed in Step 1.

3) Preprocessing of Target Character Areas (Original Character Coverage):

The area to be replaced is determined according to the coordinate information of the target character in the original contract image. A white rectangle is used to cover the original character area, which eliminates the interference of the original character on the subsequent superposition of deformed characters and ensures the background consistency of the replacement area.

4) Accurate Superposition and Replacement of Deformed Character Images:

The deformed character images loaded in Step 2 are scaled proportionally, and the transparent channel is retained. According to the coordinate position of the target character in the original image, the deformed character images are accurately superimposed onto the preprocessed area, completing the automatic replacement of target characters in the contract text (local details are shown in Figure 14).

5) Verification of Replacement Effect via Subjective and Objective Indicators: The effect of the replaced contract text is verified. From the subjective perspective, it is judged through naked-eye observation that the visual difference between the contract after embedding and the original contract is very small. From the objective perspective, the Structural Similarity Index (SSIM) and Peak Signal-to-Noise Ratio (PSNR) are calculated. The results show that the SSIM reaches 0.9872 and the PSNR reaches 27.03 dB, verifying the high concealment and excellent image quality of the replacement effect.

鉴于乙方在文化传播项目开展过程，有技术人才短缺的问题。按照按本协议约定为乙方提供相关服务。

甲方将为乙方员工提供技术培训课程，涵盖技术基础知识讲解实操指导及项目经验分享等。协议旨在提升一方员工技术能力。

培训服务费用总额为伍万元整，合同协议签订后五个工作日，乙方支付贰万元作为预付款，培训的课自完成一半，乙方支付贰万元剩余一万元与培训结束后十个工作日内支付，协议的支付方式为银行转账。

Figure 14. Watermark contract detail image

4.3 Recognition of "Dot" Stroke Deformed Characters

1) Preparation of Experimental Data and Templates:

Prepare the watermark contract images to be evaluated. Meanwhile, group the character template images corresponding to the four groups of binary subcodes ("00", "01", "10", "11") to construct a benchmark dataset for matching judgment.

2) ROI Cropping of Target Areas:

Based on the coordinate information of the replaced deformed characters in the contract text, accurately crop the Region of Interest (ROI) of the corresponding target characters on the watermark contract image to ensure the accuracy of the areas to be matched.

3) Image Preprocessing (Elimination of Noise and Illumination Interference):

Perform preprocessing on the cropped ROI and each group of character template images respectively: first conduct grayscale normalization to unify the grayscale range, then perform adaptive binarization to highlight character outlines, and finally carry out histogram equalization to enhance contrast. This process eliminates the interference of noise and illumination differences on the matching results.

4) Calculation of Quantitative Indicators for Matching Degree:

For each pair of preprocessed ROI and template images, calculate two matching degree indicators: Structural Similarity Index (SSIM), which is used to measure the consistency of structural information between the two; and Peak Signal-to-Noise Ratio (PSNR), which is used to evaluate the degree of difference in image quality.

5) Judgment of Matching Results:

Set the judgment threshold for successful matching: when SSIM is no less than 0.9 and PSNR is no less than 20 dB, the ROI is judged to match the corresponding template successfully; otherwise, the matching is judged to fail.

6) Evaluation of Recognition Effect and Result Output:

Organize the matching results and output an evaluation table (as shown in Table 7) that includes the original character, expected binary code, SSIM value, PSNR value, and comprehensive matching result. This table is used to quantitatively analyze the recognition effect of the algorithm.

Table 7. recognition result

Original character	Expected binarg code	SSIM	PSNR	Combined results
协	01	0.9791450936637309	22.181028178233042	Success
协	01	0.9791450936637309	22.181028178233042	Success
协	01	0.9791450936637309	22.181028178233042	Success
协	01	0.9791450936637309	22.181028178233042	Success
协	01	0.9791450936637309	22.181028178233042	Success
协	01	0.9848861390896512	24.152769442655323	Success
协	01	0.9791450936637309	22.181028178233042	Success
协	01	0.9791450936637309	22.181028178233042	Success
协	01	0.9791450936637309	22.181028178233042	Success
协	01	0.9791450936637309	22.181028178233042	Success
协	01	0.9791450936637309	22.181028178233042	Success
协	01	0.9791450936637309	22.181028178233042	Success
协	01	0.9791450936637309	22.181028178233042	Success
约	10	0.9766042093991316	22.181028178233042	Success
约	10	0.9766042093991316	22.181028178233042	Success
约	10	0.9766042093991316	22.181028178233042	Success
约	10	0.9766042093991316	22.181028178233042	Success
约	10	0.9766042093991316	22.181028178233042	Success
约	10	0.9766042093991316	22.181028178233042	Success
约	10	0.9766042093991316	22.181028178233042	Success
约	10	0.9766042093991316	22.181028178233042	Success
约	10	0.9766042093991316	22.181028178233042	Success
约	10	0.9766042093991316	22.181028178233042	Success
约	10	0.9766042093991316	22.181028178233042	Success
约	10	0.9766042093991316	22.181028178233042	Success
约	10	0.9766042093991316	22.181028178233042	Success
约	10	0.9766042093991316	22.181028178233042	Success
的	00	0.9712468183731374	22.181028178233042	Success

的	00	0.9712468183731374	22.181028178233042	Success
的	00	0.9712468183731374	22.181028178233042	Success
的	00	0.9712468183731374	22.181028178233042	Success
的	00	0.9976207643007555	29.79548374704095	Success
的	00	0.9712468183731374	22.181028178233042	Success
的	00	0.9712468183731374	22.181028178233042	Success
的	00	0.9712468183731374	22.181028178233042	Success
的	00	0.9712468183731374	22.181028178233042	Success
付	11	0.9828075960028921	24.07159054043353	Success
付	11	0.9828075960028921	24.07159054043353	Success
付	11	0.9828075960028921	24.07159054043353	Success
付	11	0.9828075960028921	24.07159054043353	Success
付	11	0.9828075960028921	24.07159054043353	Success
付	11	0.9923140887494681	27.576996250877386	Success
付	11	0.9828075960028921	24.07159054043353	Success
Average		0.9782	22.82	

As can be seen from the above data, when matching characters in the watermark contract with the character library, the average Structural Similarity Index (SSIM) is 97.82% and the average Peak Signal-to-Noise Ratio (PSNR) is 22.82 dB, both meeting the criteria for successful matching.

5. Conclusion

Aiming at the field of text anti-counterfeiting and focusing on the problem of complex design and generation of Chinese characters, this study proposes a method for generating "dot" stroke Chinese deformed characters based on Conditional Generative Adversarial Network (CGAN). The study constructs a dataset containing 524 high-frequency "dot" stroke Chinese characters and their corresponding deformed characters using the edge detection method, and designs a CGAN model composed of a U-Net generator and a lightweight discriminator to realize the conversion from normal characters to "dot" stroke deformed characters. Verification results show that the average SSIM between the deformed characters generated by the model and manually designed samples reaches 97.13%, and the PSNR exceeds 20.26 dB; after replacing the contract text, the average SSIM for recognition is 97.82% and the PSNR is 22.82 dB, which is better than traditional methods. This study provides an efficient and high-precision technical solution for text anti-counterfeiting, and has theoretical and practical reference value.

References

- [1] ZHU Q Q. Research on Text Steganography Based on Chinese Character Stroke Fing-Tuning[D].Hangzhou Dianzi University, 2023.
- [2] LIU D, CHEN S, ZHOU M T. Text Digital Watermarking Technology Based on Topology of Alphabetic Symbol[J]. Journal of Chinese Computer Systems, 2007, (05): 812-815.
- [3] Chang J, Gu Y. Chinese typography transfer[J]. arXiv preprint arXiv: 1707. 04904, 2017.

- [4] Isola P, Zhu J Y, Zhou T, et al. Image-to-image translation with conditional adversarial networks[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2017: 1125-1134.
- [5] Xiao C, Zhang C, Zheng C. Fontcode: Embedding information in text documents using glyph perturbation[J]. ACM Transactions on Graphics (TOG), 2018, 37(2): 1-16.
- [6] SUN S, ZHANG W M, FANG H, et al. Automatic generation of Chinese document watermarking fonts[J]. Journal of Image and Graphics, 2022, 27(01): 262-276.
- [7] Teng Shaohua, Kong Lingrui. Chinese Font Style Transfer Based on Generative Adversarial Networks[J]. Application Research of Computers, 2019, 36(10): 3164-3167. DOI: 10.19734/j.issn.1001-3695.2018.04.0330
- [8] Bai Haijuan, Zhou Wei, Wang Cunrui, et al. Font Style Transfer Method Based on Generative Adversarial Networks[J]. Journal of Dalian Minzu University, 2019, 21(03): 250-256. DOI: 10.13744/j.cnki.cn21-1431/g4.2019.03.012.
- [9] XIANG T, YANG Q H, DENG Y, et al. Combining Canny edge detection algorithm and k-means algorithm for image text recognition[J]. Neijiang Science and Technology Journal, 2020, 41(10): 39-41+61.
- [10] GUO S L, PIAO Z J. Statistical Report on the Number of Strokes in the GB13000.1 Character Set: Chinese Character Order (Stroke Order) Specification[J]. Modern Chinese, 2006, (11): 39-40.