Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2024.052611

ARTICLE

EDU-GAN: Edge Enhancement Generative Adversarial Networks with Dual-Domain Discriminators for Inscription Images Denoising

Yunjing Liu[1,#], Erhu Zhang[1,2,#,*], Jingjing Wang[3], Guangfeng Lin[2] and Jinghong Duan[4]

[1]School of Mechanical and Precision Instrument Engineering, Xi'an University of Technology, Xi'an, 710048, China

[2]Department of Information Science, Xi'an University of Technology, Xi'an, 710054, China

[3]School of Faculty of Painting, Packaging Engineering and Digital Media, Xi'an University of Technology, Xi'an, 710048, China

[4]School of Computer Science and Engineering, Xi'an University of Technology, Xi'an, 710048, China

*Corresponding Author: Erhu Zhang. Email: eh-zhang@xaut.edu.cn

#Yunjing Liu and Erhu Zhang are co-first authors with equal contributions

Received: 08 April 2024 Accepted: 12 June 2024 Published: 18 July 2024

ABSTRACT

Recovering high-quality inscription images from unknown and complex inscription noisy images is a challenging research issue. Different from natural images, character images pay more attention to stroke information. However, existing models mainly consider pixel-level information while ignoring structural information of the character, such as its edge and glyph, resulting in reconstructed images with mottled local structure and character damage. To solve these problems, we propose a novel generative adversarial network (GAN) framework based on an edge-guided generator and a discriminator constructed by a dual-domain U-Net framework, i.e., EDU-GAN. Unlike existing frameworks, the generator introduces the edge extraction module, guiding it into the denoising process through the attention mechanism, which maintains the edge detail of the restored inscription image. Moreover, a dual-domain U-Net-based discriminator is proposed to learn the global and local discrepancy between the denoised and the label images in both image and morphological domains, which is helpful to blind denoising tasks. The proposed dual-domain discriminator and generator for adversarial training can reduce local artifacts and keep the denoised character structure intact. Due to the lack of a real-inscription image, we built the real-inscription dataset to provide an effective benchmark for studying inscription image denoising. The experimental results show the superiority of our method both in the synthetic and real-inscription datasets.

KEYWORDS

Dual-domain discriminators; inscription images; denoising; edge-guided generator

1 Introduction

Inscription images have a high value and irreplaceable role, but mess noise commonly exists in real-inscription images, hindering reading and understanding. Cleanly inscribed images are essential for influencing the performance of downstream tasks, such as character recognition [1], font style transfer [2], and other high-level computer vision tasks. Therefore, it is a crucial step to remove the noise from the inscription images.

Edge prior is an essential component of the image feature, which provides more texture and detailed information in the reconstruction tasks of natural images [3,4]. Meanwhile, a fine edge can provide localization information for segmentation and object detection. Both of the mentioned properties of edge prior are helpful to the task of reconstructing high-quality inscription images. Another important prior is skeleton prior, which provides rich topological information for an object [5] and maintains semantic coherence and structural features [6]. As a result, the edge and glyph structure of the character should be preserved correctly in the inscription image denoising task. On the contrary, missing the two glyph information will result in restored images with blurred local and incoherent font structures. Fig. 1 exemplifies the inscription images denoising task, where (a) represents the noisy image. (b and c) represent the results of incorrect denoising, where (b) has severe artifacts and (c) has damaged font structures, and we highlight the inaccurate parts with red boxes. (d) represents the results of correct denoising.

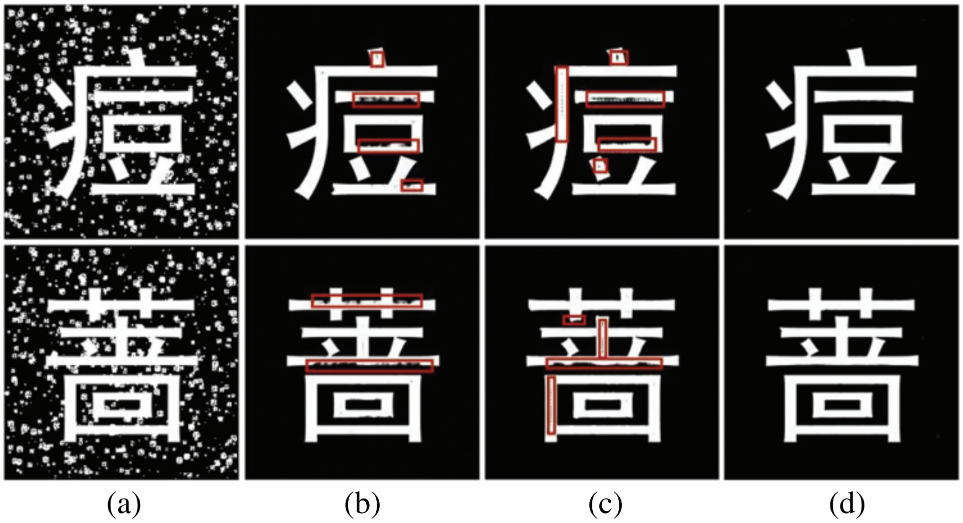

Figure 1: Inscription images denoising examples. (a) Noisy image (b and c) Incorrect denoising results, marking the inaccurate parts of font structure with red boxes, (d) Accurate denoising results

Recently, some inscription image restoration methods have been studied by establishing noise patterns to remove noise. For example, Feng [7] proposed a Gaussian mixture method to simulate noise in document images, but it is unsuitable for complex real-inscription image degradations. Subsequently, Zhang et al. designed a more complex noise model [8,9], where they modeled the noise of documents and inscriptions by adding dots and squares to clean images. Zhang et al. [10] and Wang et al. [11] introduced a GAN to perform blind inscription image denoising, improving image restoration performance. However, inscription images with mottled local and damaged font structures since these works mainly consider pixel-level image restoration and ignore glyph information.

We propose an EDU-GAN framework to address the mentioned issues. The main contributions of our work are summarized as follows:

(1) Unlikely existing denoising architectures, we design edge extraction and edge guidance modules as generator conditions, making the recovered image edge consistent and improving the quality of the restoration image.
(2) We adopt morphological domain training, which boosts the complete semantics of the recovered glyph structure.

(3) Experimental results show that the EDU-GAN model is more suitable for inscription font generation and denoising.

2 Related Works

2.1 Inscription Image Restoration

Early research on inscription images mainly focused on physics-based algorithms and filters [12,13]. Recently, deep learning models have been gradually introduced to inscription image restoration. Zhang et al. [9] modeled noises in calligraphic images and then used an adversarial network to restore degraded document images. Miao et al. [8] applied simulated noise methods [9] to the inscription images. Yue et al. [14] introduced a dual adversarial network [10] to perform blind inscription image denoising. A study [15] proposed an improved autoencoder for denoising letter images in scanned invoices. Shi et al. [16] integrated skeleton information into the denoising task and introduced a global-local feature interaction module, thereby improving the denoising performance of the network. Reference [17] input the skeleton and degraded character image into the network and then used the GAN network to reconstruct the character image. However, the above works ignore the structural characteristics of Chinese character images, resulting in unsatisfactory recovery results.

2.2 GAN-Based Image Restoration

Recently, GAN networks have gradually been applied to restore clean images from degraded images. Chen et al. [18] designed the GAN network to generate paired datasets and introduced Convolutional Neural Networks for blind denoising. Yue et al. [14] proposed the simultaneous removal and generation of noise tasks in a unified Bayesian framework. Wang et al. [19] explored a more effective rainy image generation method under the Bayesian framework to improve rain removal performance. However, the above study ignores the consideration of the discriminator. A pioneer work [20] proposed a discriminator based on the U-Net architecture that simultaneously captured global and local information from images, thus encouraging the generator to construct high-quality samples. Subsequently, Huang et al. [21] applied the discriminator of the U-Net structure to medical image denoising. Wei et al. [22] introduced the discriminator of U-Net structure into the super-resolution tasks to generate more high-resolution images with better realistic. The successful application of the above method inspired us to employ based on GAN ideas for restoring inscription images.

3 The Proposed Method

Based on the above consideration, we design an EDU-GAN network for inscription image denoising, as shown in Fig. 2, which contains a generator and two based U-Net discriminators. We detail the design of network architecture and modules in this section.

3.1 Overall Framework

As shown in Fig. 2, the denoising process is to guide an edge-guided generator (EGG) to map the input noisy image $I_N \in R^{w\times h\times c}$ with size $w \times h$ and channel c into the clean inscription image $I_C \in R^{w\times h\times c}$. The process can be expressed as:

$$I_{DN} = \mathrm{G}\left(I_N\right) \approx I_C \tag{1}$$

where I_{DN} represents the reconstructed clear inscription image. The I_{DN} is input to the discriminator of the image and morphological domains, which contains an encoder-decoder structure in each discriminator. For the image domain (D^{img}), the encoder is D^{img}_{enc}, and the decoder is D^{img}_{dec}. Similarly, the encoder is D_{enc}^{mog}, and the decoder is D_{dec}^{mog} in the morphological branch (D^{mog}). The discrimination fed back the generated image and skeleton information, prompting the generator to restore more accurate images I_{DN}.

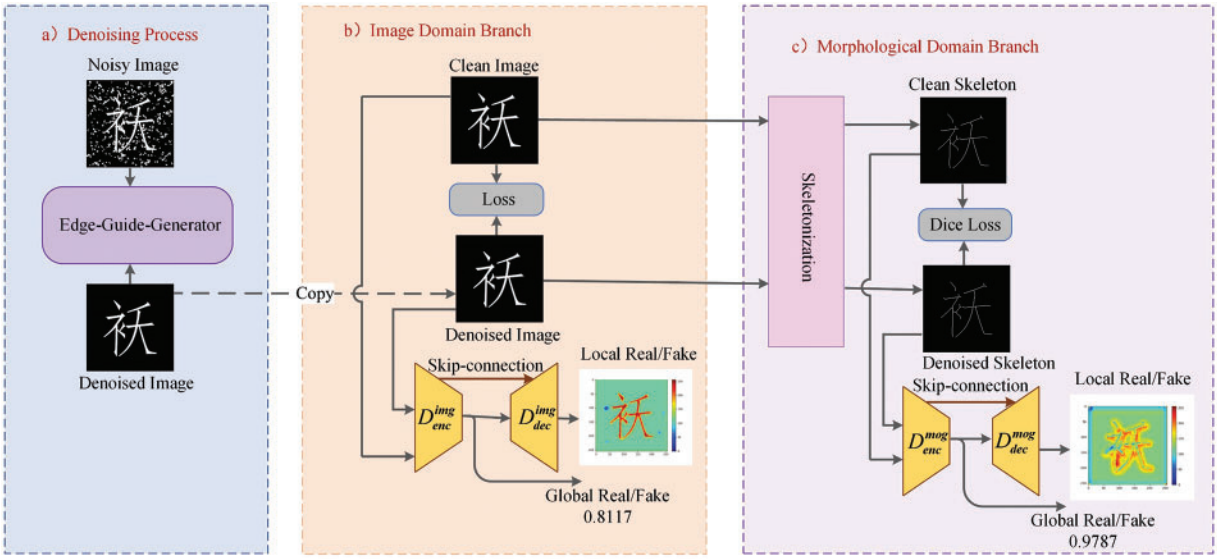

Figure 2: Inscription-denoising network framework

3.2 Generator Overview

The proposed edge-guided generator, as shown in Fig. 3. Firstly, the network performs five stages of feature extraction. Secondly, an edge extraction module (EEM) excavates edge semantics from the lowest scale-ensemble block (SEB1) and highest scale-ensemble block (SEB5). Thirdly, the asymmetric feature fusion module (AFF) is applied to enhance the information flow of different stages. Fourthly, the edge-guide feature module (EFM) integrates the font edge and contextual feature information to improve the restored text edge of consistency. Finally, multiple context prediction modules (CPM) are employed to fuse features in a bottom-up aggregate manner. The CPM1 module outputs the generator prediction results.

3.2.1 Feature Extraction Module

Considering the unknown and complexity of the noise pattern (such as the shape and size of the noise patches) on the inscription images, we design a scale-ensemble block (SEB) as shown in Fig. 4. By utilizing receptive field information of different scales to boost feature extraction capability. Specifically, the feature extraction module contains five SEB modules. SEB1 applies two 3×3 convolutions for extracting the shallow information of the inscription image, and SEB2$\sim$SEB5 concatenates three different scale convolution features and further adopts the convolution layer to model the connected features, thereby obtaining the depth features from the inscription image. The output is SEB_i^{out} $(i = 1, 2, \ldots, 5)$, which will be entered into the next SEB module. The output size of each SEB module is $\{1,\ 1/2,\ 1/4,\ 1/8,\ 1/16\}$ of the original image size. Refer to Table 1 for the detailed parameters of the SEB module.

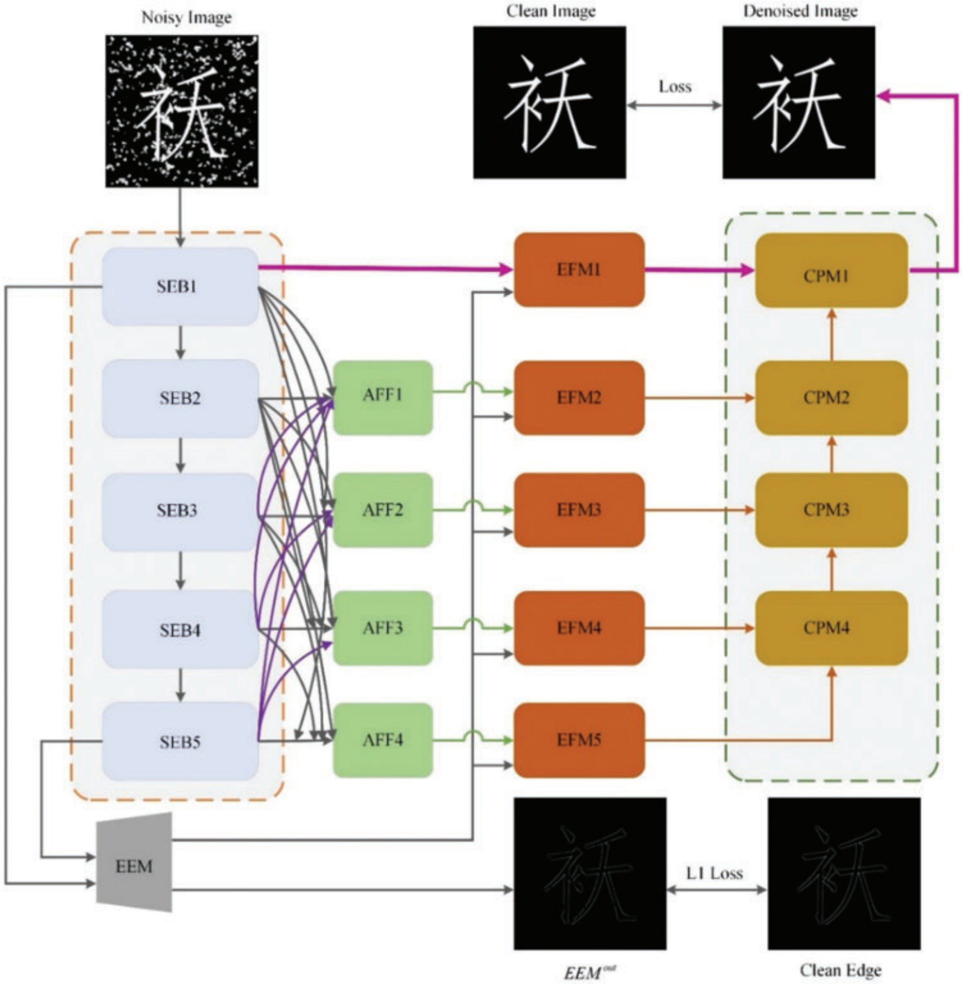

Figure 3: Edge-guide generator framework

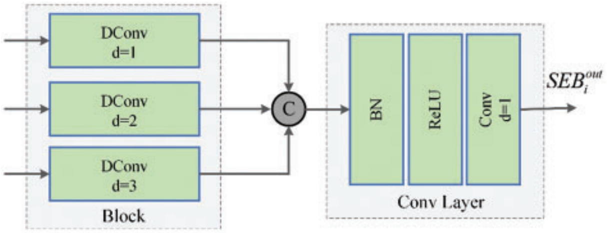

Figure 4: Scale-ensemble block

Table 1: Scale-ensemble blocks details

Stage	Parameters
SEB1	Conv3 × 3, dilation = 1, C = 64
	Conv3 × 3, dilation = 1, C = 64
SEB2	DConv: BN+ReLU+ Conv3 × 3, dilation = 1, C = 32
	DConv: BN+ReLU+ Conv3 × 3, dilation = 2, C = 32
	DConv: BN+ReLU+ Conv3 × 3, dilation = 3, C = 32
	Conv Layer: BN+ReLU+Conv4 × 4, dilation = 1, stride = 2, C = 128
SEB3	DConv: BN+ReLU+ Conv3 × 3, dilation = 1, C = 32
	DConv: BN+ReLU+ Conv3 × 3, dilation = 2, C = 32
	DConv: BN+ReLU+ Conv3 × 3, dilation = 3, C = 32
	Conv Layer: BN+ReLU+Conv4 × 4, dilation = 1, stride = 2, C = 256
SEB4	DConv: BN+ReLU+ Conv3 × 3, dilation = 1, C = 32
	DConv: BN+ReLU+ Conv3 × 3, dilation = 2, C = 32
	DConv: BN+ReLU+ Conv3 × 3, dilation = 3, C = 32
	Conv Layer: BN+ReLU+Conv4 × 4, dilation = 1, stride = 2, C = 512
SEB5	DConv: BN+ReLU+ Conv3 × 3, dilation = 1, C = 32
	DConv: BN+ReLU+ Conv3 × 3, dilation = 2, C = 32
	DConv: BN+ReLU+ Conv3 × 3, dilation = 3, C = 32
	Conv Layer: BN+ReLU+Conv4 × 4, dilation = 1, stride = 2, C = 1024

3.2.2 Edge Extraction Module

As we discussed, edges are beneficial to image reconstruction tasks, and how to extract clean and accurate text edges from noisy inscription images is crucial. As shown in Fig. 5, we build an edge extraction module (EEM) to excavate edge information from low-level features (SEB_1^{out}), which provides rich edge detail information, and high-level semantic features (SEB_5^{out}), which contain text positions. The output is EEM^{out}, which can be denoted as:

$$\begin{aligned} SEB_1^{out'} &= Conv_{1\times 1}\left(SEB_1^{out}\right) \\ SEB_5^{out'} &= Conv_{1\times 1}\left(SEB_5^{out}\right) \\ EEM^{out} &= \left[Conv_{1\times 1}\left(Conv_{3\times 3}\left(Conv_{3\times 3}\left(SEB_1^{out'} + \left(SEB_5^{out'}\uparrow\right)\right)\right)\right)\right] \end{aligned} \tag{2}$$

where ↑ represents up-sampling.

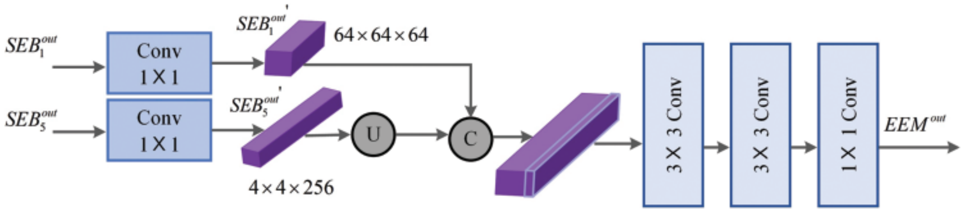

Figure 5: EEM framework

3.3 Asymmetric Feature Fusion

Due to the unknown noise level of inscription images, the feature of small noise patches is easy to obtain in the shallower layer, but large noise patches are easy to extract in the deeper layer. To further fully exploit captured shallow and deep features, we design an asymmetric feature fusion module (AFF), inspired by Cho et al. [23], which integrates features at different levels into the original features to enhance the information flow of different levels, refer to Fig. 6.

$$\begin{aligned} AFF_1^{out} &= AFF1\left(\left(SEB_1^{out}\right)^{\downarrow}, SEB_2^{out}, \left(SEB_3^{out}\right)^{\uparrow}, \left(SEB_4^{out}\right)^{\uparrow}, \left(SEB_5^{out}\right)^{\uparrow}\right) \\ AFF_2^{out} &= AFF2\left(\left(SEB_1^{out}\right)^{\downarrow}, \left(SEB_2^{out}\right)^{\downarrow}, SEB_3^{out}, \left(SEB_4^{out}\right)^{\uparrow}, \left(SEB_5^{out}\right)^{\uparrow}\right) \\ AFF_3^{out} &= AFF3\left(\left(SEB_1^{out}\right)^{\downarrow}, \left(SEB_2^{out}\right)^{\downarrow}, \left(SEB_3^{out}\right)^{\downarrow}, SEB_4^{out}, \left(SEB_5^{out}\right)^{\uparrow}\right) \\ AFF_4^{out} &= AFF4\left(\left(SEB_1^{out}\right)^{\downarrow}, \left(SEB_2^{out}\right)^{\downarrow}, \left(SEB_3^{out}\right)^{\downarrow}, \left(SEB_4^{out}\right)^{\downarrow}, SEB_5^{out}\right) \end{aligned} \quad (3)$$

where AFF_j^{out} represents the outputs of the j asymmetric feature fusion module. ↓, ↑ represent down-sampling and up-sampling.

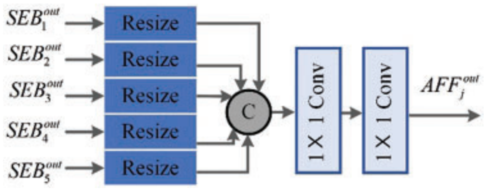

Figure 6: AFF framework

3.4 Edge-Guide Feature Module

Since the high-frequency edge features will gradually disappear with the depth of the network increase, to fully utilize the edge information to reconstruct high-quality images, we design an edge-guide feature module (EFM). The module combines the aggregated information obtained by the AFF module and edge information to supplement the feature representation of structural semantics.

As shown in Fig. 7, given a feature SEB_1^{out} or multi-scale feature AFF_j^{out} $(j = 1, 2 \ldots 4)$and an edge feature EEM^{out}, take the input AFF_j^{out} and EEM^{out} as examples. The EEM^{out} performs 1×1 convolution and Sigmoid layer, which changes from 3 channels to 1 channel. Then, AFF_j^{out} performs element-wise multiplication with $EEM^{out'}$ and element-wise addition with skip-connection. Finally, it applies 3×3 convolutions to obtain the initial feature fusion $AFF_j^{out'}$. The output of the EFM module can be described as:

$$\begin{aligned} EEM^{out'} &= D\left(\left(\sigma\left(Conv_{1\times1}\left(EEM^{out}\right)\right)\right)\right) \\ AFF_j^{out'} &= Conv_{3\times3}\left(\left(AFF_j^{out} \otimes EEM^{out'}\right) \oplus AFF_j^{out}\right) \\ EFM_i^{out} &= Conv_{1\times1}\left(\sigma\left(Conv1d\left(GAP\left(AFF_j^{out'}\right)\right)\right) \otimes AFF_j^{out'}\right) \end{aligned} \quad (4)$$

where D represents down-sampling. ⊗ and ⊕ denote element-wise product and element-wise addition, respectively.

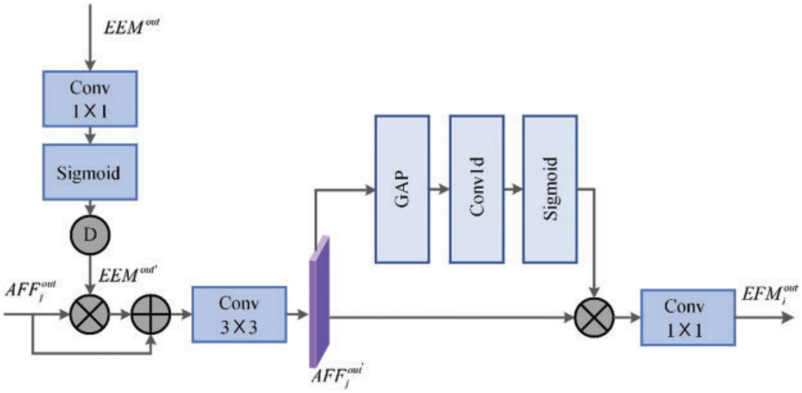

Figure 7: EFM framework

3.5 *Context Prediction Module*

As shown in Fig. 8, we design a bottom-up context prediction module (CPM) to fuse EFM_i^{out} of each layer, the CPM module output at each stage is CPM_j^{out} $(j = 1, 2...4)$, and CPM1 outputs the final prediction result CPM_1^{out}. The CPM module used similar ensemble blocks of the SEB module, especially the [IN+ReLU] layer added to the deconvolution layer in the transpose convolution scale ensemble block, and the output is $EFM_i^{out'}$. The $EFM_i^{out'}$ performs a concatenation operation with the output EFM_{i-1}^{out} of the previous EFM and then applies two 3 × 3 convolutions. We introduce an attention module (CBAM) [24] after the 3 × 3 convolution to enhance the representation ability by focusing on critical features and suppressing unnecessary information.

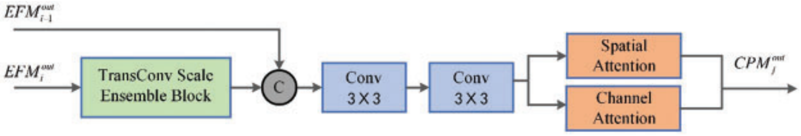

Figure 8: CPM framework

3.6 *Dual-Domain U-Net Discriminator*

3.6.1 *Image Domain U-Net Discriminator*

Inspired by [22], we introduce a U-Net architecture of the discriminator to improve the ability of GAN networks to represent global and local differences. The discriminator consists of an encoder module, a decoder module, and several skip connections attention modules. Encoder D_{enc}^{img} focuses on learning global structural information in the image domain. The decoder D_{dec}^{img} is responsible for capturing the difference in local information between real and fake samples.

3.6.2 *Morphological Domain U-Net Discriminator*

The noise may damage the font structure in the inscriptions. To ensure the complete semantics of the font structure, we propose branch of the morphological domain to maintain the skeleton from the inscription image reconstruction. Specifically, the denoised image first inputs the morphological operation of skeleton extraction, i.e., $I_{DSK} = SK(I_{DN}) \approx SK(I_C) = I_{CK}$, and then is input to the morphological domain discriminator D^{mog}. Among them, D^{mog} and D^{img} adopt the same network architecture. Likewise, the encoder D^{mog} of D^{mog}_{enc} captures global information, the decoder D^{mog}_{dec} captures local information, SK represents the skeleton extraction operation, I_{DSK} represents the denoised image skeleton, and I_{CK} represents the clean inscription image skeleton.

3.7 Loss Function

3.7.1 Generator Loss

The generator adopts compound loss, including content loss, perceptual loss, and adversarial loss, with different perspectives to better constrain and encourage the generator, which produces better realistic images. The detailed losses are as follows:

Adversarial Loss: We employ the Least Squares GAN [25] as discriminator loss. The total adversarial loss can be expressed as:

$$L_{adv} = \lambda_{im\_do} \underbrace{\left[E_{I_{DN}} \left[D^{img}_{enc} \left(I_{DN} \right) - 1 \right]^2 + E_{I_{DN}} \left[D^{img}_{dec} \left(I_{DN} \right) - 1 \right]^2 \right]}_{\text{image domain}} + \lambda_{mo\_do} \underbrace{\left[E_{I_{DN}} \left[D^{img}_{enc} \left(I_{DSK} \right) - 1 \right]^2 + E_{I_{DN}} \left[D^{img}_{dec} \left(I_{DSK} \right) - 1 \right]^2 \right]}_{\text{morphological domain}} \tag{5}$$

where λ_{im_do} and λ_{mo_do} are the weights for the adversarial loss in the image and the morphological domain, respectively. We set weights to 0.005 and 0.01.

Content Loss: The image reconstruction loss L_{img}, edge reconstruction loss L_{edge}, and skeleton reconstruction loss L_{sk} constitute content loss L_{count}, among L_{img} and L_{edge} introduce pixel loss, and L_{sk} uses dice loss [26]. It can be formulated as:

$$L_{img} = E_{\left(I_{DN}, I_{\text{C}} \right)} \left(|| I_{DN} - I_C ||^2 \right) \tag{6}$$

$$L_{edge} = E_{\left(I_{DE}, I_{\text{CE}} \right)} \left(|| I_{DE} - I_{CE} ||_1 \right) \tag{7}$$

$$L_{sk} = 1 - 2 \frac{\sum_i I_{DSK_i} I_{CK_i}}{\sum_i \left(I_{DSK_i} \right)^2 + \sum_i \left(I_{CK_i} \right)^2} \tag{8}$$

$$L_{count} = \lambda_{img} L_{img} + \lambda_{edge} L_{edge} + \lambda_{sk} L_{sk} \tag{9}$$

where λ_{img}, λ_{edge} and λ_{sk} are the weights for L_{img}, L_{edge} and L_{sk}, we set weights to 1, 0.5 and 0.5, respectively.

Feature Loss: We also consider the feature-level loss L_{feat}, employing the pre-trained model VGG19 [27] $VGG\,(.)$ on the ImageNet dataset to extract I_{DN} and I_C features. L_{feat} contains perceptual loss L_{perc}, style loss L_{style} [28], and contrastive loss L_{cr} [29], as follows:

$$L_{perc} = E_{\left(I_C, I_{DN} \right)} \left[|| VGG\left(I_C \right) - VGG\left(I_{DN} \right) ||_1 \right] \tag{10}$$

$$L_{style} = E_{\left(I_C, I_{DN} \right)} \left[|| \gamma \left(VGG\left(I_C \right) \right) - \gamma \left(VGG\left(I_{DN} \right) \right) ||_1 \right] \tag{11}$$

$$L_{cr} = \frac{|| VGG\left(I_C \right) - VGG\left(I_{DN} \right) ||_1}{|| VGG\left(I_N \right) - VGG\left(I_{DN} \right) ||_1} \tag{12}$$

$$L_{feat} = \beta_{perc} L_{perc} + \beta_{style} L_{style} + \beta_{cr} L_{cr} \tag{13}$$

where γ (.) denotes the Gram matrix operation. β_{perc}, β_{style}, and β_{cr} are the weights of L_{perc}, L_{style}, and L_{cr}, respectively. We set weights to 0.01, 120 and 0.1. Finally, the total loss L_G of the generator can be expressed as follows:

$$L_G = L_{adv} + L_{count} + L_{feat} \tag{14}$$

3.7.2 Discriminator Loss

The discriminator loss L_D, including discriminative loss $L_{D^{img}}$ and skeleton discriminative loss $L_{D^{mog}}$, from the image and morphological branches, respectively. Based on the least-square GAN background, D^{img} and D^{mog} can be formulated as:

$$\begin{aligned} L_{D^{img}} = {} & \underbrace{E_{I_C}\left[D^{img}_{enc}\left(I_C\right) - 1\right]^2 + E_{I_{DN}}\left[D^{img}_{enc}\left(I_{DN}\right)\right]^2}_{\text{global adversarial}} \\ & + \underbrace{E_{I_C}\left[D^{img}_{dec}\left(I_C\right) - 1\right]^2 + E_{I_{DN}}\left[D^{img}_{dec}\left(I_{DN}\right)\right]^2}_{\text{local adversarial}} \end{aligned} \tag{15}$$

$$\begin{aligned} L_{D^{mog}} = {} & \underbrace{E_{I_C}\left[D^{mog}_{enc}\left(I_{CK}\right) - 1\right]^2 + E_{I_{DSK}}\left[D^{mog}_{enc}\left(I_{DSK}\right)\right]^2}_{\text{global adversarial}} \\ & + \underbrace{E_{I_C}[D^{mog}_{dec}\left(I_{CK}\right) - 1]^2 + E_{I_{DSK}}[D^{mog}_{dec}\left(I_{DSK}\right)]^2}_{\text{local adversarial}} \end{aligned} \tag{16}$$

Therefore, the total loss for the discriminator is:

$$L_D = L_{D^{img}} + L_{D^{mog}} \tag{17}$$

4 Experiments

4.1 Training Details

The proposed method of EDU-GAN is implemented by PyTorch in the PyCharm integrated development environment and conducted on NVIDIA GeForce RTX 4070 with 12 GB GPU. During training, the initial learning rates of the discriminators D^{img}, D^{mog} and the generator EGG are 0.004, 0.004, and 0.002, respectively. The learning rates are from 200 iterations to begin decay, which is half every 100 iterations. The number of iterations and the batch size N_b are 500 and 18, respectively. In each iteration, we randomly crop noisy inscription images I_N of 18 × 100 patches with size 64 × 64 in the D_{data} to train the network. In addition, the entire framework of EDU-GAN in Fig. 2 is updated using the Adam optimizer.

4.1.1 Algorithm

Algorithm 1 details the EDU-GAN network implementation process. The network inputs the noisy inscription image I_N and the clean image I_C, which is optimized and trained for 500 iterations, and the network outputs the denoised image I_{DN}.

Algorithm 1: Pseudo-code of EDU-GAN method.

```
Input: Training data $D_{data} = \left\{I_{N_n}, I_{C_n}\right\}_{n=1}^{N}$, batch size $N_b$, number of modules $i = \{1, 2, \ldots, 5\}$ and $j = \{1, 2, \ldots, 4\}$, $iter = 500$.
Output: the final denoised inscription image $I_{DN}$.
    Randomly initialize the model parameter $\theta$.
    For $m = 1$ to $iter$, do
        $\{I_N, I_C\} \leftarrow \text{Sample}\ (D_{date}, N_b)$.
        $SEB_i^{out} \leftarrow \text{SEB}_i\ (I_N)$.
        $AFF_j^{out} \leftarrow \text{AFF}_j\left(SEB_1^{out}, SEB_2^{out}, SEB_3^{out}, SEB_4^{out}, SEB_5^{out}\right)$.
        $EEM^{out} \leftarrow \text{EEM}\left(SEB_1^{out}, SEB_5^{out}\right)$.
        $EFM_i^{out} \leftarrow \text{EFM}_i\left(SEB_1^{\text{out}} \text{ or } AFF_j^{out}, EEM^{out}\right)$.
        $I_{DN} \leftarrow \text{CPM}_j\left(EFM_i^{out}, EFM_{i-1}^{out}\right)$.
        $I_{DSK} \leftarrow SK\ (I_{DN})$.
        $I_{CK} \leftarrow SK\ (I_C)$.
        Update EGG with fixed $D^{img}$, $D^{mog}$.
        Update $D^{img}$, with fixed EGG, $D^{mog}$.
        Update $D^{mog}$, with fixed EGG, $D^{img}$.
    End for
```

4.2 Dataset Description

In our EDU-GAN, we require noisy Chinese characters images with stains or scratches to train our network, but those kinds of public datasets are lacking. Therefore, we create synthetic inscription and real-inscription datasets to evaluate the performance of the EDU-GAN network. Our synthetic datasets are constructed by adding noise patches on widely used printed Chinese character images[1] and handwritten dataset[2]. And the real-inscription dataset is collected from websites[3].

4.2.1 Synthetic Inscription Datasets

Printed Chinese Character Datasets: We generate 41,305 images, including 11 fonts (such as FangSong). Each font contains 3755 commonly used Chinese characters in GB2312. We use Random Walk Modeling (RWM) and the Square Circle Noise Modeling method [9], adding noise to the printed Chinese character images to create synthetic datasets, denoting as D_{syn} and D_{scm}, respectively. We use 33,044 images for training and 8261 for testing.

Handwritten Chinese Character Dataset: HWDB1.1 is a handwritten Chinese character dataset containing 1,176,000 images from 300 writers. We select 65,709 images written by 30 people, which are preprocessing and adding noise by RWM methods to create synthetic dataset, denoted as D_{hw}. This dataset uses 52,568 for training and 13,141 for testing.

4.2.2 Real-Inscription Dataset

Due to the lack of a real-inscription image dataset, we collect inscription images from different dynasties. These images are cropped into 1356 images with 256 × 256 size and manually processed to

[1] https://www.foundertype.com (accessed on 10/05/2024)
[2] http://www.nlpr.ia.ac.cn/databases/handwriting/Download.html (accessed on 10/05/2024)
[3] https://www.9610.com/index.htm (accessed on 10/05/2024)

produce corresponding clean inscription images. We build the real-inscription images dataset[4], denoted as D_{real}, including 1,085 inscription images for training and 271 for testing. This dataset provides an effective benchmark for studying inscription images denoising and is available within the article.

4.3 Evaluation Metrics

There are two commonly used metrics to evaluate the performance in image reconstruction, i.e., peak signal-to-noise ratio (PSNR) and structural similarity (SSIM). However, there is no quantitative index for font structure in inscription denoising, so we propose to evaluate the similarity between skeletons, i.e., SGap.

The main idea of SGap is that the skeleton of the denoised and clean images is the same or similar. Specifically, given a noisy image I_N, input it into the trained network to get I_{DN}, and then through the morphological skeleton extraction operation to obtain I_{DSK}, which can be expressed as:

$$SGap = SSIM\left(SK\left(I_{DN}\right)\right) - SSIM\left(SK\left(I_{N}\right)\right) \tag{18}$$

Obviously, the larger the value of the first term on the right side of the equation, the $I_{DSK} = SK\left(I_{DN}\right)$ and I_{CK} are closer, and the larger the SGap.

4.4 Experiment Results and Analysis

We compare the performance of the EDU-GAN network with other classic image restoration methods, including calligraphy image denoising CIDGan [9], CharFormer [16], and RCRN [17], considering edge priors denoising methods EdCNN [30] and Mlefng [4] and the well-performing image restoration methods DnCNN [31], Uformer [32], VRGNet [19] and the established and representative methods CBDNet [33] and NBNet [34] for natural image denoising. To fairly evaluate the performance of different methods by setting and providing training environment and data to be the same.

4.4.1 Comparison with Other Methods

Results on Synthetic Datasets: In Table 2, we summarize the performance of our EDU-GAN and other methods on the D_{syn},D_{scm} and D_{hw}. Our EDU-GAN achieves better performance than other methods. For intuitive demonstration, we present the visualization of denoised images on the D_{scm} dataset. As shown in Fig. 9, all methods can remove noise patches, but these methods generally have artifacts and severe edge distortion (marked by red arrows). Compared with other methods, the proposed method can more effectively preserve and restore the edge details of Chinese characters in the image, indicating that our network is more suitable for inscription image denoising. Moreover, we have visualized the output of each part of the network. Taking the EDU-GAN network trained on D_{syn} as an example, the trained model can be directly used for noisy inscription images in the test dataset and visualize the output of the network component, as shown in Fig. 10. The edge extraction module (EEM) can predict the fine edges from the noisy image in the test dataset, the encoder D_{enc}^{img} of the discriminator D^{img} outputs the global score, and the decoder D_{dec}^{img} outputs the confidence map and focuses on the text area. Similarly, the encoder D_{enc}^{mog} of the discriminator D^{mog} outputs the global score, and the decoder D_{dec}^{mog} focuses on font structure.

[4] https://github.com/liuyunjing0306/EDU-GAN (accessed on 10/05/2024)

Table 2: Performance comparisons among different methods on the synthetic and real-inscription datasets

Methods	D_{scm}			D_{syn}			D_{hw}			D_{real}		
	PSNR	SSIM	SGap	PSNR	SSIM	SGap	PSNR	SSIM	SGap	PSNR	SSIM	SGap
Raw image	9.167	0.3528	–	9.410	0.3903	–	9.156	0.3598	–	16.701	0.7472	–
DnCNN	36.680	0.9822	0.5978	32.774	0.9554	0.5500	26.036	0.9773	0.5839	23.860	0.8973	0.1720
CBDNet	35.011	0.9874	0.5905	32.991	0.9104	0.5464	26.353	0.9598	0.5852	23.886	0.9335	0.1714
NBNet	35.068	0.9914	0.5945	32.380	0.9553	0.5474	25.847	0.9449	0.5867	23.379	0.9525	0.1721
Uformer	27.140	0.9698	0.5650	19.485	0.8782	0.4696	20.689	0.9451	0.5542	21.372	0.9284	0.1612
VRGNet	33.422	0.9835	0.5945	30.279	0.8939	0.5360	26.184	0.8574	0.5845	21.415	0.8499	0.1511
EdCNN	32.498	0.9845	0.5856	32.168	0.9523	0.5422	26.261	0.9757	0.5831	22.745	0.9251	0.1596
Mlefng	32.871	0.9870	0.5922	29.009	0.9086	0.5269	24.179	0.9559	0.5758	21.422	0.9132	0.1512
CIDGan	35.300	0.9908	0.5960	32.345	0.9567	0.5441	25.433	0.9793	0.5841	21.943	0.9341	0.1610
CharFormer	33.847	0.9839	0.5907	30.502	0.9621	0.5468	26.135	**0.9824**	0.5862	23.404	0.9411	0.1692
RCRN	33.994	0.9866	0.5912	29.963	0.9392	0.5323	24.624	0.9737	0.5783	19.314	0.8921	0.1268
EDU-GAN	**37.880**	**0.9916**	**0.6002**	**34.260**	**0.9623**	**0.5502**	**26.386**	0.9812	**0.5874**	**24.204**	**0.9563**	**0.1723**

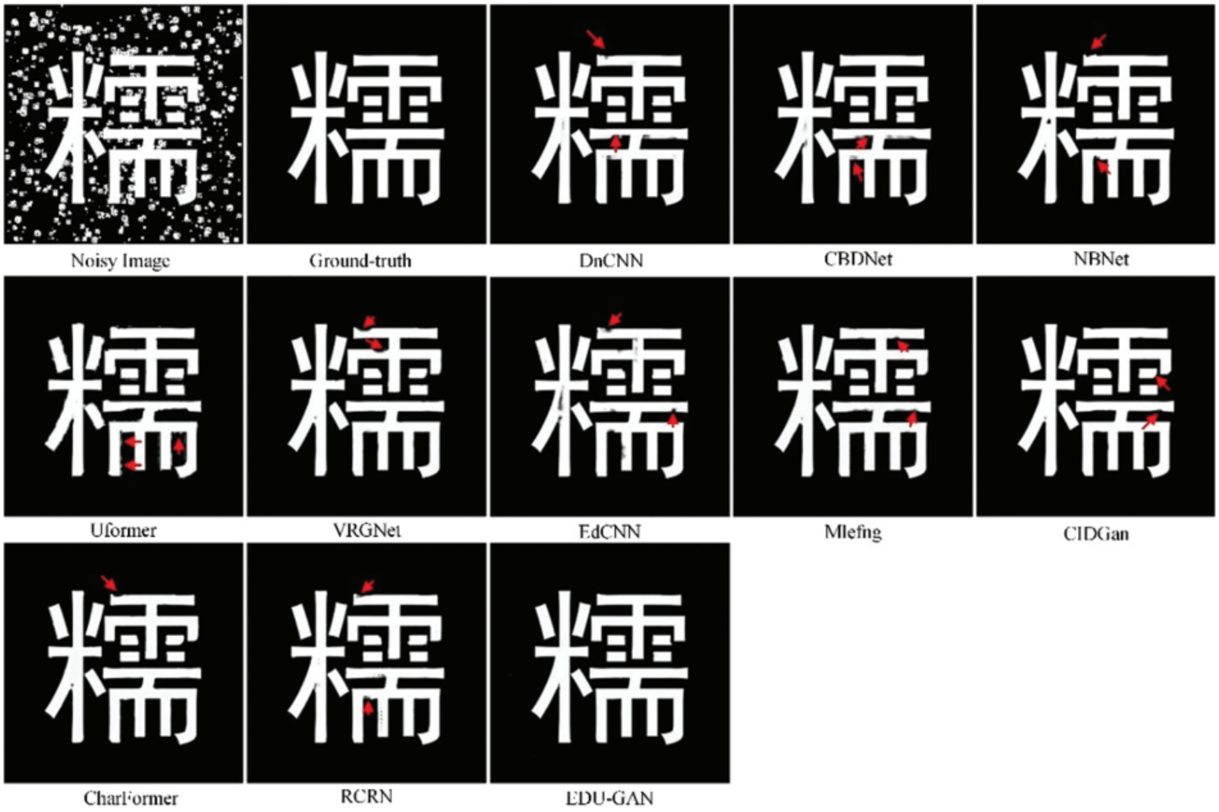

Figure 9: Visualization of denoising results on the D_{scm}

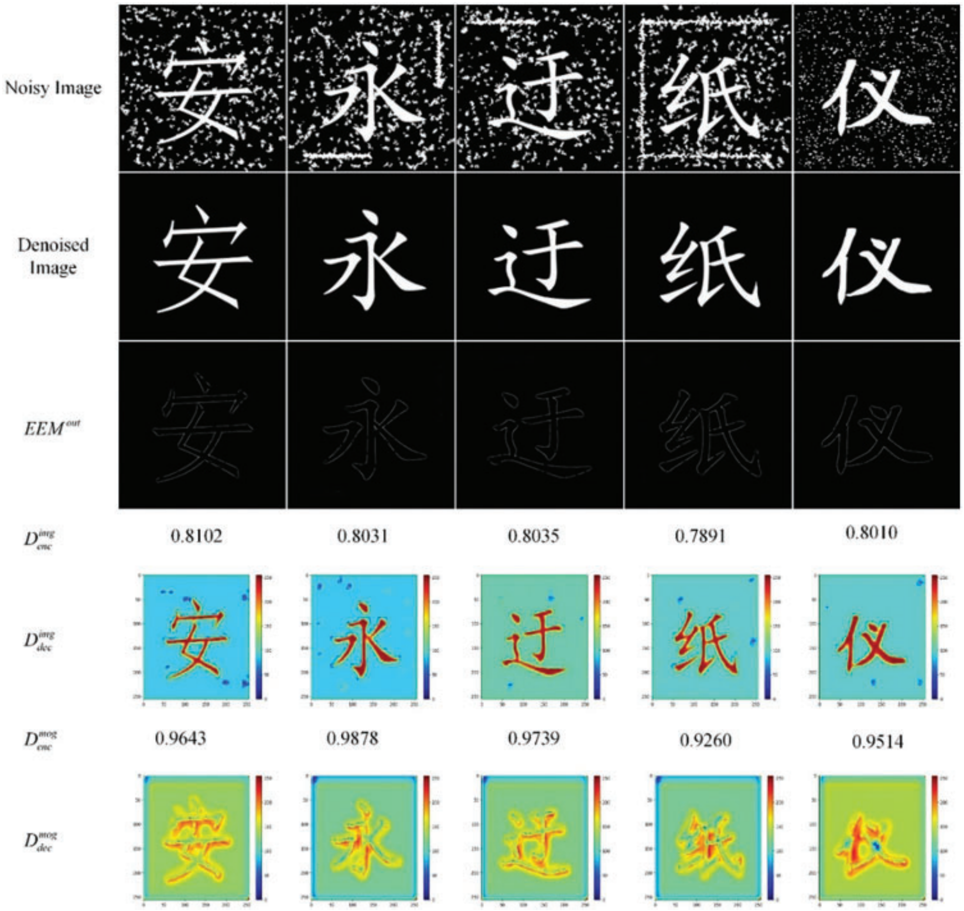

Figure 10: Visualization of different parts of the EDU-GAN trained on the D_{syn}. Note that blue and red indicate lower and higher confidence scores, respectively

Results on Real-Inscription Dataset: As shown in Table 2, we also compare EDU-GAN with other methods on the D_{real} dataset. Compared with synthetic datasets, the real-inscription images are more complex degradation and result in removing noise difficulty. Our EDU-GAN outperforms all other methods with 24.204 dB PSNR and 0.9563 SSIM on D_{real}. We also compare EDU-GAN with other methods on the quality of denoising results, provided in Fig. 11. We can observe that DnCNN, CBDNet, VRGNet, EdCNN, Mlefng, and RCRN cannot successfully remove dense noise patches and suffer from the gray-mottled artifacts (as shown in the red marked area). In addition, NBNet has a more serious phenomenon of destroying the font component structure. CIDGan, CharFormer and Uformer achieve the restoration with higher quality. However, they still cannot thoroughly remove the noise in some regions (as shown in the green rectangles). Our method, EDU-GAN, recovers the best image quality, and the exact font integrity is closest to the ground truth image.

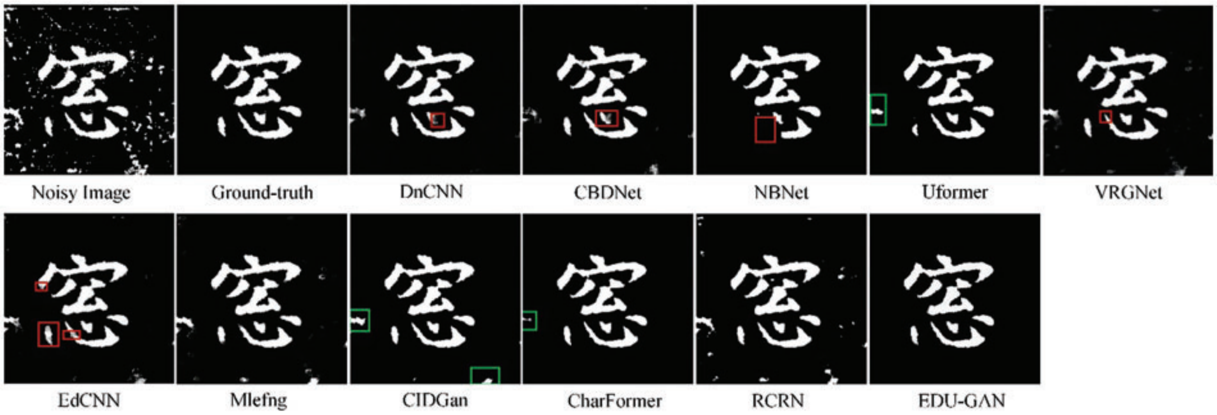

Figure 11: Visualization of real-inscription denoising results

4.4.2 Ablative Study

Components Analysis: Table 3 presents the quantitative results of the ablation studies. To verify each design in EDU-GAN is reasonable. First, benefiting from the discriminative architecture that provides global and local information for the generator, EDU-GAN with two U-Net discriminators achieves the highest PSNR and SSIM than EDU-GAN-D^{img} and EDU-GAN-D^{img}-D^{mog}. Specifically, the PSNR, SSIM, and SGap of EDU-GAN with dual domains are 1.095, 0.0137, and 0.0003 higher than those without dual domains (EDU-GAN-D^{img}-D^{mog}), respectively. Second, ablation experiments of each component in the generator. Compared with EDU-GAN-D^{img}-D^{mog}, EDU-GAN-D^{img}-D^{mog}-AFF-EEM-EFM with PSNR, SSIM, and SGap reduced by 4.481, 0.0077, and 0.0009, respectively. It denotes that the generator (EGG) with AFF, EEM, and EFM modules can effectively retain the key features of the text, making the generator more accurate in removing noise. In addition, the performance with the attention mechanism increased by 0.501, 0.048, and 0.0273, respectively, compared to the performance without the attention mechanism, indicating that the attention mechanism can focus on critical features, thereby enhancing the network's denoising performance.

Table 3: Ablation study results on different modules

Methods	PSNR	SSIM	SGap
EDU-GAN	**34.260**	**0.9623**	**0.5502**
EDU-GAN-D^{img}	34.162	0.9598	0.5497
EDU-GAN-D^{img}-D^{mog}	33.165	0.9486	0.5499
EDU-GAN-D^{img}-D^{mog}-AFF	32.807	0.9441	0.5495
EDU-GAN-D^{img}-D^{mog}-EEM-EFM	31.474	0.9315	0.5459
EDU-GAN-D^{img}-D^{mog}-AFF-EEM-EFM	28.684	0.9409	0.5490
EDU-GAN-D^{img}-D^{mog}-AFF-EEM-EFM- CBAM	28.183	0.8929	0.5217

Architectures of Discriminator: Since the discriminator is crucial and cannot be ignored in the GAN network, it is worth exploring the highlights of the U-Net structure discriminator compared with other classic discriminators, such as the patch discriminator [35], global discriminator [36], pixel discriminator [35]. For fair comparisons, using the same overall pipeline, referred to as Fig. 2, the generator uses the EGG network, and the discriminator applies the above three classic discriminators, respectively. Table 4 reports the assembly of global and local information in the discriminator of the U-Net structure achieved the best PSNR and SSIM scores, which denotes our U-Net discriminator outperforms the other three mentioned classical discriminators for inscription image denoising. Because the classic discriminator can only focus on local or global information, it cannot better represent the differences between images. Unlike classical discriminators, the U-Net constructed discriminator can simultaneously capture global and local features in images, thereby guiding the generator towards restoring better images and improving the denoising performance of the network.

Table 4: Ablation study results on different discriminators

Methods	PSNR	SSIM	SGap
Patch	32.998	0.9107	0.5413
Global	31.827	0.9549	0.5471
Pixel	32.556	0.9426	0.5477
U-Net	**34.260**	**0.9623**	**0.5502**

Effect of Patch Size: The selection of patch size is also important during the training process, so we randomly cut the training data set into 64 × 64, 128 × 128, and 256 × 256 pixels to train our model. Table 5 shows that small patches can achieve better denoising performance, which is likely because smaller patches can better capture local contextual information. Additionally, larger patches increase computational complexity and may result in insufficient training samples, ultimately leading to ineffective model training. Therefore, smaller patches can attain better denoising performance. To be a more visual representation, we provide the denoising results of different patch sizes in Fig. 12. As shown, the image quality restored by smaller patches has the highest quality, and the recovered characters are closest to the ground truth image.

Table 5 : Ablation study results on different patch sizes

Patch sizes	PSNR	SSIM	SGap
64 × 64	**34.260**	**0.9623**	**0.5502**
128 × 128	31.234	0.9569	0.5451
256 × 256	30.171	0.9213	0.5267

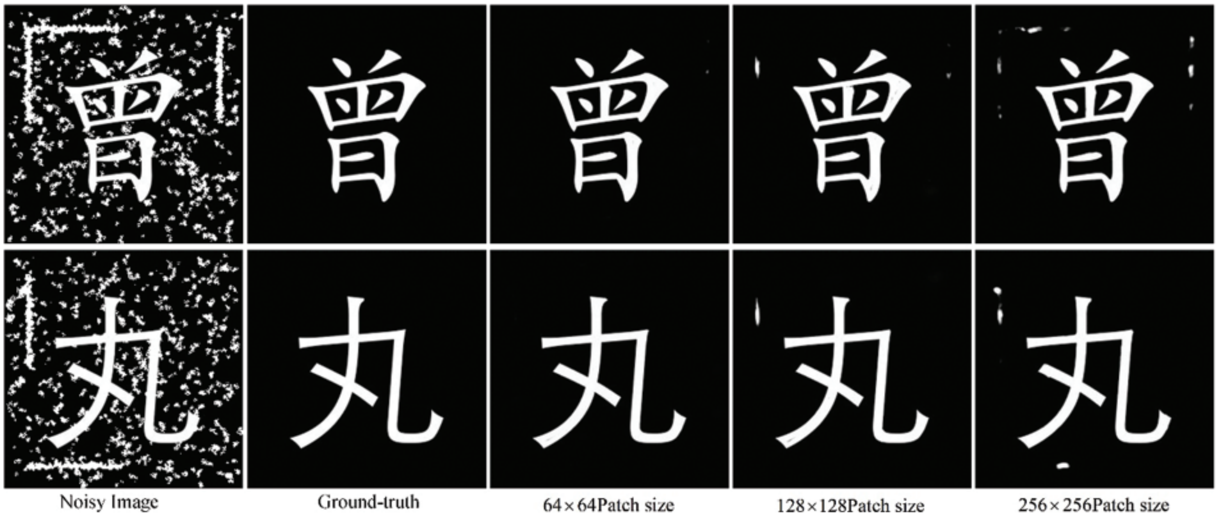

Figure 12: Visualization of denoising results at patch sizes

Effect of Loss Functions: The loss of our generator has three parts, which encourages the generator to produce more high-precision denoised images. We compare the impact of denoising with/without these losses, as shown in Table 6. These results on D_{syn} indicate that the loss function with capturing feature-level information loss (L_{feat}) and the skeleton loss (L_{sk}) function obtains the best performance.

Table 6: Ablation study results on loss functions

Methods	L_G	$L_G - L_{cr}$	$L_G - L_{cr} - L_{perc}$	$L_G - \underbrace{L_{cr} - L_{perc} - L_{style}}_{L_{feat}}$	$L_G - L_{feat} - L_{sk}$
PSNR	**34.260**	32.948	32.459	30.171	29.285
SSIM	**0.9623**	0.9415	0.9589	0.9380	0.9340
SGap	**0.5502**	0.5436	0.5475	0.5286	0.5296

Network Complexity Analysis: Table 7 provides the model parameters, training time, and inference time of different methods. From Table 7, it shows that more complex models require longer training times. Although our model parameters and training time are not advantageous, it achieves the best denoising performance. In particular, our network application scenarios do not require real-time, so we are more tolerant of network training time and inference time. For limited devices, our model needs to be further lightweight.

Table 7: Network complexity analysis. Note: Model parameters are in units of M, and time is in units of s

Methods	Parameters	Training time	Inference time
DnCNN	0.5583	7792.9215	0.0054
CBDNet	4.3654	10,459.6548	0.0065

(Continued)

Table 7 (continued)

Methods	Parameters	Training time	Inference time
NBNet	10.4552	16,030.6980	0.0332
Uformer	556.4987	24,252.6255	0.1807
VRGNet	6.3610	12,382.5443	0.0150
EdCNN	0.0820	3788.5015	0.0029
Mlefng	6.8599	14,118.4067	0.0646
CIDGan	0.9426	8978.4516	0.0161
CharFormer	13.2260	20,609.8823	0.0745
RCRN	11.0130	16,596.9125	0.0087
EDU-GAN	170.8706	23,253.9459	0.0305

Generalization Analysis: Table 8 shows that different methods are trained on three independent synthetic datasets and then directly used for testing on real-inscription images to verify the robustness ability of the network. Our EDU-GAN achieves better robustness performance than other methods from Table 8. Taking the EDU-GAN network trained on D_{hw} as an example, our method improves PSNR and SSIM by 3.294 and 0.1071, respectively, compared to the lowest values. Specifically, the ability to remove noise from real- inscriptions is more significant on D_{syn} and D_{hw}, which may be related to the amount of data and the method of adding noise when creating the synthetic dataset.

Table 8: The network generalization analysis. Different methods are trained on a synthetic dataset (D_{scm}/D_{syn}/D_{hw}) and tested on the same real inscription images (D_{real})

Methods	D_{scm}			D_{syn}			D_{hw}		
	PSNR	SSIM	SGap	PSNR	SSIM	SGap	PSNR	SSIM	SGap
DnCNN	17.680	0.7558	0.1169	18.310	0.7693	0.1149	20.701	0.8145	0.1537
CBDNet	17.867	0.7590	0.1151	17.823	0.7490	0.1154	20.699	0.8144	0.1539
NBNet	18.593	0.7790	**0.1226**	18.457	0.7778	0.1202	21.606	0.8391	**0.1636**
Uformer	16.405	0.7493	0.1180	18.228	**0.8073**	0.1217	18.560	0.8895	0.1402
VRGNet	16.327	0.7735	0.1204	18.094	0.7900	0.1214	20.995	0.8305	0.1557
EdCNN	18.368	**0.7993**	0.1176	18.385	0.7843	0.1233	20.945	0.9142	0.1517
Mlefng	17.948	0.7520	0.1178	18.076	0.7916	0.1191	20.684	0.9143	0.1570
CIDGan	16.861	0.7801	0.1150	16.898	0.7639	0.1239	20.375	0.9183	0.1557
CharFormer	17.222	0.7596	0.1161	17.626	0.7691	0.1245	21.589	0.9298	0.1633
RCRN	16.295	0.7479	0.1140	17.472	0.7483	0.1024	20.374	0.9204	0.1520
EDU-GAN	**18.838**	0.7882	0.1185	**18.491**	0.7923	**0.1246**	**21.854**	**0.9215**	0.1634

5 Discussion and Conclusion

In the paper, we propose EDU-GAN, a novel framework to remove noise from inscription images. In the generator, the scale-ensemble block (SEB) can model the multi-scale features and improve the ability to detect noise patterns with unknown and complexity. The asymmetric feature fusion module (AFF) is beneficial for obtaining rich multi-level features by effectively aggregating the features of multiple SEB modules. The edge-guide feature module (EFM) is responsible for effectively integrating edge information from the edge extraction module (EEM) and context information to enhance the representation ability of edge information in the deep network. The bottom-up content prediction module (CPM) fuses the features between layers and outputs the prediction results. In the discriminator, we introduce U-Net architecture, providing global and local information for the generator and further applying it to the morphological domain, which enhances the skeleton features and structural integrity of fonts. We comprehensively evaluate the performance of EDU-GAN on synthetic and real-inscription datasets, which demonstrates superior performance gains over other methods.

Although our network can obtain high evaluation indicators, it still has some limitations. Firstly, the successful application of our network is crucial to multiple downstream tasks such as font style recognition and font restoration. However, the size of our network model is relatively large, such as the SEB module. Existing computing resources can afford the proposed network, but our model needs to be further lightweight for devices with limited memory. Secondly, our network is also applicable to non-experts. When the network is trained, the noisy inscription image is input into the network, and the network outputs the denoised image. However, an intuitive user interface may be easier to operate for non-experts, which is a part that needs improvement for future deployment and implementation.

In conclusion, our model performs well in denoising inscription images and has good generalization ability, but the model size is large and needs to be further lightweight in the future.

Acknowledgement: The authors sincerely appreciate the reviewers and the editorial team for their valuable feedback and scientific suggestions.

Funding Statement: This work was supported by the Key R&D Program of Shaanxi Province, China (Grant Nos. 2022GY-274, 2023-YBSF-505); the National Natural Science Foundation of China (Grant No. 62273273).

Author Contributions: Erhu Zhang: Supervision, Project administration, Methodology, Writing–review & editing. Yunjing Liu: Data curation, Methodology, Software, Validation, Writing–original draft. Jingjing Wang: Data curation Guangfeng Lin: Data curation. Jinghong Duan: Data curation. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The authors confirm that the data supporting the findings of this study are available within the article, which can be found at the link below.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] R. Mithe, S. Indalkar, and N. Divekar, "Optical character recognition," *Int. J. Recent Technol. Eng.*, vol. 2, no. 1, pp. 72–75, 2013.

[2] Y. Gao and J. Wu, "Gan-based unpaired chinese character image translation via skeleton transformation and stroke rendering," in *Proc. 34th AAAI Conf*, New York, NY, USA, 2020, pp. 646–653. doi: 10.1609/aaai.v34i01.5405.

[3] S. Chupraphawan and C. A. Ratanamahatana, “Deep convolutional neural network with edge feature for image denoising,” in *Proc. Adv. Intell. Sys. Comput.*, Bangkok, Thailand, 2020, pp. 169–179. doi: 10.1007/978-3-030-19861-9_17.

[4] F. Fang, J. Li, Y. Yuan, T. Zeng, and G. Zhang, “Multilevel edge features guided network for image denoising,” *IEEE Trans. Neural Networks Learn. Sys.*, vol. 32, no. 9, pp. 3956–3970, 2020. doi: 10.1109/TNNLS.2020.3016321.

[5] C. Liu, Y. Tian, Z. Chen, J. Jiao, and Q. Ye, “Adaptive linear span network for object skeleton detection,” *IEEE Trans. Image Process*, vol. 30, pp. 5096–5108, 2021. doi: 10.1109/TIP.2021.3078079.

[6] J. Cai, L. Peng, Y. Tang, C. Liu, and P. Li, “TH-GAN: Generative adversarial network based transfer learning for historical Chinese character recognition,” in *Proc. Int. Conf. Doc. Anal. Recognit.*, Sydney, NSW, Australia, IEEE, 2019, pp. 178–183. doi: 10.1109/ICDAR.2019.00037.

[7] S. Feng, “A novel variational model for noise robust document image binarization,” *Neurocomputing*, vol. 325, no. 1, pp. 288–302, 2019. doi: 10.1016/j.neucom.2018.09.087.

[8] Y. Miao, L. Li, Y. Ji, and G. Li, “Research on denoising method of chinese ancient character image based on chinese character writing standard model,” *Sci. Rep.*, vol. 12, no. 1, pp. 19795, 2022. doi: 10.1038/s41598-022-24388-y.

[9] J. Zhang, M. Guo, and J. Fan, “A novel generative adversarial net for calligraphic tablet images denoising,” *Multimedia Tools Appl.*, vol. 79, no. 1–2, pp. 119–140, 2020. doi: 10.1007/s11042-019-08052-8.

[10] H. Zhang, Y. Qi, X. Xue, and Y. Nan, “Ancient stone inscription image denoising and inpainting methods based on deep neural networks,” *Discret. Dyn. Nat. Soc.*, vol. 2021, pp. 1–11, 2021. doi: 10.1155/2021/7675611.

[11] X. Wang, K. Wu, Y. Zhang, Y. Xiao, and P. Xu, “A gan-based denoising method for chinese stele and rubbing calligraphic image,” *Visual Comput.*, vol. 39, no. 4, pp. 1351–1362, 2023. doi: 10.1007/s00371-022-02410-8.

[12] F. Ge and L. He, “A de-noising method based on L0 gradient minimization and guided filter for ancient Chinese calligraphy works on steles,” *Eurasip J. Image Video Process.*, vol. 2019, no. 1, pp. 32, 2019. doi: 10.1186/s13640-019-0423-x.

[13] Z. Shi, B. Xu, X. Zheng, and M. Zhao, “An integrated method for ancient Chinese tablet images de-noising based on assemble of multiple image smoothing filters,” *Multimedia Tools Appl.*, vol. 75, no. 19, pp. 12245–12261, 2016. doi: 10.1007/s11042-016-3421-3.

[14] Z. Yue, Q. Zhao, L. Zhang, and D. Meng, “Dual adversarial network: Toward real-world noise removal and noise generation,” in *ECCV 2020: 16th Euro. Conf.*, Glasgow, UK, 2020, vol. 12355, pp. 41–58. doi: 10.1007/978-3-030-58607-2_3.

[15] S. I. Alshathri, D. J. Vincent, and V. S. Hari, “Denoising letter images from scanned invoices using stacked autoencoders,” *Comput. Mater. Contin.*, vol. 71, no. 1, pp. 1371–1386, 2022. doi: 10.32604/cmc.2022.022458.

[16] D. Shi *et al.*, “CharFormer: A glyph fusion based attentive framework for high-precision character image denoising,” in *Proc. 30th ACM Int. Conf. Multimed.*, 2022, pp. 1147–1155. doi: 10.1145/3503161.3548208.

[17] D. Shi, X. Diao, H. Tang, X. Li, H. Xing and H. Xu, “RCRN: Real-world character image restoration network via skeleton extraction,” in *Proc. 30th ACM Int. Conf. Multimed.*, 2022, pp. 1177–1185. doi: 10.1145/3503161.3548344.

[18] J. Chen, J. Chen, H. Chao, and M. Yang, “Image blind denoising with generative adversarial network based noise modeling,” in *2018 IEEE CVPR*, Salt Lake City, UT, USA, 2018, pp. 3155–3164. doi: 10.1109/CVPR.2018.00333.

[19] H. Wang, Z. Yue, Q. Xie, Q. Zhao, Y. Zheng and D. Meng, “From rain generation to rain removal,” in *2021 IEEE CVPR*, Nashville, TN, USA, 2021, pp. 14786–14796. doi: 10.1109/CVPR46437.2021.01455.

[20] E. Schonfeld, B. Schiele, and A. Khoreva, “A U-Net based discriminator for generative adversarial networks,” in *2020 IEEE CVPR*, Seattle, WA, USA, 2020, pp. 8207–8216. doi: 10.1109/CVPR42600.2020.00823.

[21] Z. Huang, J. Zhang, Y. Zhang, and H. Shan, "DU-GAN: Generative adversarial networks with dual-domain U-Net-based discriminators for low-dose CT denoising," *IEEE Trans. Instrum. Meas.*, vol. 71, pp. 1–12, 2021. doi: 10.1109/TIM.2021.3128703.

[22] Z. Wei, Y. Huang, Y. Chen, C. Zheng, and J. Gao, "A-ESRGAN: Training real-world blind super-resolution with attention U-Net discriminators," in *20th Pacific Rim Int. Confer. Artif Intell. (PRICAI)*, Jakarta, Indonesia, 2023, vol. 14327, pp. 16–27. doi: 10.1007/978-981-99-7025-4_2.

[23] S. J. Cho, S. W. Ji, J. P. Hong, S. W. Jung, and S. J. Ko, "Rethinking coarse-to-fine approach in single image deblurring," in *Proc. IEEE Int. Conf. Comput. Vision*, Montreal, QC, Canada, 2021, pp. 4621–4630. doi: 10.1109/ICCV48922.2021.00460.

[24] S. Woo, J. Park, J. Y. Lee, and I. S. Kweon, "CBAM: Convolutional block attention module," in *ECCV 2018: 15th Euro. Conf.*, Munich, Germany, 2018, vol. 12211, pp. 3–19. doi: 10.1007/978-3-030-01234-2_1.

[25] X. Mao, Q. Li, H. Xie, R. Y. Lau, Z. Wang, and S. P. Smolley, "Least squares generative adversarial networks," in *Proc. IEEE Int. Conf. Comput. Vision*, Venice, Italy, 2017, pp. 2813–2821. doi: 10.1109/ICCV.2017.304.

[26] N. H. Nguyen, "U-Net based skeletonization and bag of tricks," in *Proc. IEEE Int. Conf. Comput. Vision*, Montreal, BC, Canada, 2021, pp. 2105–2109. doi: 10.1109/ICCVW54120.2021.00238.

[27] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *3rd Int. Conf. Learn. Represent.*, San Diego, CA, USA, 2015.

[28] C. Liu, Y. Liu, L. Jin, S. Zhang, C. Luo and Y. Wang, "EraseNet: End-to-end text removal in the wild," *IEEE Trans. Image Process.*, vol. 29, pp. 8760–8775, 2020. doi: 10.1109/TIP.2020.3018859.

[29] H. Wu *et al.*, "Contrastive learning for compact single image dehazing," in *2021 IEEE CVPR*, Nashville, TN, USA, 2021, pp. 10546–10555. doi: 10.1109/CVPR46437.2021.01041.

[30] T. Liang, Y. Jin, Y. Li, and T. Wang, "EDCNN: Edge enhancement-based densely connected network with compound loss for low-dose CT denoising," in *15th IEEE Int. Conf. Signal Process. Proc.*, Beijing, China, 2020, vol. 2020, pp. 193–198. doi: 10.1109/ICSP48669.2020.9320928.

[31] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising," *IEEE Trans. Image Process.*, vol. 26, no. 7, pp. 3142–3155, 2017. doi: 10.1109/TIP.2017.2662206.

[32] Z. Wang, X. Cun, J. Bao, W. Zhou, J. Liu and H. Li, "Uformer: A general u-shaped transformer for image restoration," in *2022 IEEE CVPR*, New Orleans, LA, USA, 2022, pp. 17662–17672. doi: 10.1109/CVPR52688.2022.01716.

[33] S. Guo, Z. Yan, K. Zhang, W. Zuo, and L. Zhang, "Toward convolutional blind denoising of real photographs," in *2019 IEEE CVPR*, Long Beach, CA, USA, 2019, pp. 1712–1722. doi: 10.1109/CVPR.2019.00181.

[34] S. Cheng, Y. Wang, H. Huang, D. Liu, H. Fan and S. Liu, "NBNet: Noise basis learning for image denoising with subspace projection," in *2021 IEEE CVPR*, Nashville, TN, USA, 2021, pp. 4894–4904. doi: 10.1109/CVPR46437.2021.00486.

[35] J. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proc. IEEE Int. Conf. Comput. Vision*, Venice, Italy, 2017, pp. 2242–2251. doi: 10.1109/ICCV.2017.244.

[36] I. Goodfellow *et al.*, "Generative adversarial nets," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.