

Integrating Visual and Contextual Cues for Ancient Chinese Rubbing Image Restoration

HAO XIA, Shenzhen University, China

XUETING LIU, Saint Francis University, China

CHENGZE LI, Saint Francis University, China

HUISI WU, Shenzhen University, China

ZHENKUN WEN, Shenzhen University, China

TONG-YEE LEE, National Cheng Kung University, Taiwan

Rubbing served as a crucial cultural carrier, transferring characters from steles onto paper using ink, thereby facilitating greater dissemination and longer-term preservation. However, prolonged corrosion, inclement weather, and other factors cause various rubbing deteriorations. Therefore, an automated rubbing image restoration method is highly desirable to reduce the complexity of manual restoration procedures. In particular, when we refer to restoring a rubbing image, we mean recreating the original text on the rubbing image to reflect its original appearance. Existing methods, such as image denoising, can address minor deterioration but cannot handle severe deterioration. Image inpainting methods attempt to handle large areas of deterioration but generally struggle to produce contextually consistent strokes due to a lack of control in the restoration process. Additionally, currently there is no reliable method for rubbing character recognition. In this paper, we propose a novel two-stage rubbing image restoration method that performs well on different levels of deterioration. In the first stage, we integrate visual and contextual information for text recognition in rubbing images, including those that are heavily deteriorated. The second stage includes a multi-prior conditional latent diffusion model for rubbing image restoration, utilizing the character identities recognized in the first stage and undeteriorated characters as priors to guide the restoration process. Extensive experiments have demonstrated the effectiveness of our method compared to existing approaches.

CCS Concepts: • **Applied computing** → **Optical character recognition**; • **Computing methodologies** → **Image processing**; **Reconstruction**.

Additional Key Words and Phrases: Rubbing image restoration, image inpainting, image denoising, character recognition, language model, diffusion model

1 INTRODUCTION

In ancient times, carving stones was a crucial method for permanently preserving text-based messages. After the invention of paper, the text on the stones was commonly replicated onto paper by rubbing for easier dissemination. With the advent of computers, these replications are further digitized into *rubbing images* (Fig. 1 left). Unfortunately, ancient stones often have different levels of deterioration caused by corrosion, bad weather, or human damage as time passes. Restoring the original writing information of a rubbing image is important for various purposes, such as education and historical research. In particular, when we refer to restoring a rubbing image, we mean recreating the original text on the rubbing image with its original appearance. An example is shown on the right side of Fig. 1, where all deteriorated text has been restored to closely resemble the original from a human perspective. The most challenging aspect of this task is addressing the severe deterioration that obstructs the restoration of the text. Although visual-based image restoration methods, such as denoising [1–3] and inpainting [4–6] might be able to resolve slightly

H. Xia, H. Wu, and Z. Wen are with College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, Guangdong, 518060, China. E-mails: xiahao2022@email.szu.edu.cn, hswu@szu.edu.cn, wenzk@szu.edu.cn

X. Liu (corresponding author) and C. Li are with Yam Pak Charitable Foundation School of Computing and Information Sciences, Saint Francis University, Hong Kong SAR, China. E-mails: tliu@sfu.edu.hk, czli@sfu.edu.hk

T.-Y. Lee is with Dept. of Computer Science and Information Engineering, National Cheng Kung University, Tainan, Taiwan. E-mail: tonylee@mail.ncku.edu.tw.

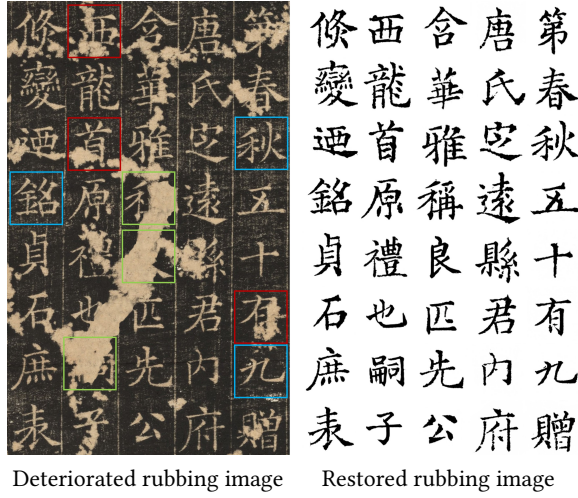


Fig. 1. Our method automatically restores rubbing images with different extents of deterioration. We classify the deteriorated characters into three categories: minor (blue boxes), intermediate (red boxes), and severe deterioration (green boxes). Credits: *Guo Yun mu zhi*. Tang Dynasty, Chinese. Source: [Chinese Rubbings Collection - CURIOSity Digital Collections](#).

deteriorated characters (the blue boxes in Fig. 1) and intermediately/partially deteriorated characters (the red boxes in Fig. 1), they inevitably fail on heavily deteriorated characters due to the corruption of visual information (the green boxes in Fig. 1).

Therefore, rubbing restoration remains a tedious manual task for professional restoration experts, consisting of two steps: recovering the context of deteriorated characters and reconstructing deterioration-free characters. Unfortunately, neither step is easy for computers, even with state-of-the-art deep learning methods. For context recovery, optical character recognition (OCR) methods (e.g. [7, 8]) can recognize characters with slight or partial deterioration, but they are inaccurate for heavily deteriorated characters. Some methods first apply visual-based OCR to recognize characters as much as possible and then use language models as post-processing to refine the results [9, 10]. However, these approaches may accumulate errors when handling long sequences, as both OCR and language models work on noisy observations. Our first challenge is to recognize characters with severe deterioration by exploring both the visual and the language domains. For character reconstruction, existing methods commonly use font generation methods [11–13]. However, these methods generally regenerate the entire character to preserve the topology, which can cause deviations from the original character, especially in the undeteriorated parts. A conditional character reconstruction method that retains the undeteriorated character parts is highly desired to reconstruct partially deteriorated characters. This is the second challenge we aim to resolve.

In this paper, we propose a novel image restoration method tailored for rubbing images. Our method addresses two challenges: 1) recognizing and restoring text in severely deteriorated rubbing images written in ancient Chinese, and 2) restoring the deteriorated rubbing image pixels with visual aesthetics and consistency with the undeteriorated parts. To achieve this, we first introduce a recognition module for recognizing text in deteriorated rubbing images. Our key argument is that visual and contextual information should be considered simultaneously. Visual information serves as the basis for context analysis, and contextual information enhances the accuracy of visual data. We propose an iterative approach that refines the recognized character identities using both visual

and contextual information, enabling the recognition of long sequences of severely deteriorated characters. Next, to reconstruct the deteriorated characters with their identities, we propose a multi-prior conditional diffusion model. This model is trained on a large and diverse set of character images covering hundreds of fonts, and performs conditional font generation based on character identities and style guidance from less deteriorated characters in the same rubbing image. We perform a wide variety of evaluations in terms of text recognition and reconstruction of ancient Chinese rubbings of different ages and fonts, demonstrating superior performance. We also validate our systematic designs with an ablation study. Moreover, we discuss the potential of applying our approach to non-Chinese languages, such as Greek and Roman steles.

Our major contributions can be summarized as follows:

- We propose a novel rubbing-tailored image restoration method that works satisfactorily on severely deteriorated rubbing images.
- We propose a few-shot method to recognize and restore text in a severely deteriorated rubbing image written in ancient Chinese.
- We propose a multi-prior conditional diffusion model to restore a deteriorated rubbing image so that the deteriorated part is restored with pleasant visual effects and the undeteriorated part is properly preserved.

2 RELATED WORK

2.1 Image Restoration

Image restoration aims to restore a deteriorated image to its original version. We roughly classify the existing restoration methods in our scope as denoising and inpainting methods.

2.1.1 Image Denoising. With the advent of deep learning, DnCNN [14] introduced a fully convolutional neural network with batch normalization for denoising. Subsequent improvements include dense networks [15], encoder-decoder architectures [16], and transformer-based methods [17–19]. However, these methods primarily address additive Gaussian noise or noise from in-camera processing in natural images, which differ from deterioration in rubbing images.

2.1.2 Image Inpainting. Inpainting methods fill larger holes than denoising methods and are able to synthesize new content in the unrecoverable image parts. The advent of generative adversarial networks [20] led to context encoders [21], which encode corrupted images into latent representations and then decode them for inpainting. Improvements such as Fourier convolutions [4] and transformer blocks [5, 22] have improved the efficiency and effectiveness of inpainting. Recently, RePaint [6] proposed guiding inpainting with a pre-trained denoising diffusion probabilistic model [23] as a generative prior. However, these methods are designed for visual consistency and cannot reconstruct the correct character topology without text recognition. In contrast, our approach incorporates character text as a prior during the restoration process, crucial for accurately reconstructing heavily deteriorated characters.

2.2 Rubbing-tailored Restoration

Image restoration methods generally fail with rubbing images for two reasons: 1) Rubbing images differ significantly from natural images, so models trained on natural images perform poorly on rubbings; 2) Without text guidance, these methods struggle to restore the character topology in rubbing images accurately. Consequently, rubbing-tailored restoration methods have been developed to address these challenges.

2.2.1 Rubbing Denoising. Several rubbing-tailored denoising methods have been proposed, employing model-based approaches with rubbing data [24, 25]. Zhang et al. [2] proposed CIDG, which

utilizes the pix2pix [26] framework with residual dense blocks to map noisy to clean rubbing images. RCRN [3] used character skeleton information to restore real-world character images using a GAN-based framework. CharFormer [27] introduced skeleton fusion and attention mechanisms to recover character images while preserving the inherent skeleton. While these methods address minor deterioration, they are insufficient for resolving intermediate and severe deterioration due to the absence of text guidance.

2.2.2 Guided Rubbing Synthesis. RubbingGAN [12] proposed a two-step approach to handle deterioration: 1) text recognition with manual correction and 2) text-based font generation. However, text recognition can only recover text with intermediate deterioration and still requires manual input for severely deteriorated characters. Additionally, the GAN-based approach for regenerating characters often generates distorted and inaccurate strokes due to the limited number of characters in rubbings and the instability of adversarial training. Yang et al. [28] proposed a diffusion-based method to predict the original appearance of damaged historical documents. However, this method still requires manual input for the text/context in the damaged region. In comparison, our method automatically recovers text even from severely deteriorated rubbings and uses a multi-prior conditional diffusion model to restore rubbings with consistent and pleasant visuals.

2.3 Text Recognition

Text recognition is also highly related to rubbing restoration, as the recognized text identity can be used as priors to synthesize deteriorated characters. Extracting text from images through visual recognition is often referred to as optical character recognition (OCR) [7, 8, 29]. Although OCR methods work well on mildly deteriorated images, they struggle with severely deteriorated characters. Some methods attempt to mitigate this problem by post-processing the top-1 OCR results using pre-trained language models to predict the identity of missing or heavily deteriorated characters from the text-domain contexts. For example, Nguyen et al. [9] detect errors with a BERT model [30] and correct them with a neural machine translation model. Soper et al. [10] proposed post-processing the OCR output with a finetuned BART [31] language model for error detection and correction. Zhu et al. [32] proposed to cross-reference the text against a corpus of ancient inscription texts to derive the output. However, using language models or cross-references as a simple add-on to OCR approaches may suffer from error accumulation, since the recognized text contexts are very unreliable prior to the language model processing steps. Some methods train visual and language models together to improve recognition accuracy [33, 34]. Vision language models (VLMs) have also been proposed for text recognition recently [35–38]. However, these models are primarily trained on natural scene text and struggle to transfer effectively to rubbing images. Furthermore, the lack of ground-truth text labeling in the rubbing images hinders the finetuning or training of these models. In contrast, we propose refining text recognition results iteratively using both visual and language models, which can improve the accuracy of text recognition without requiring explicit labeling of rubbing images. We show our experimental evaluations in Section 6.

2.4 Font Generation

Font generation synthesizes text images from the text identity along with one or several reference styles. If we manage to recover the identity of a deteriorated rubbing character, font generation can produce an image similar to the ground truth. Early font generation methods used image-to-image translation models for font style transfer [39], but these required extensive paired data. Few-shot font generation techniques overcame this by using content and style disentanglement [11, 40]. While these methods can mimic the style of undeteriorated characters, they cannot faithfully reconstruct partially deteriorated characters.

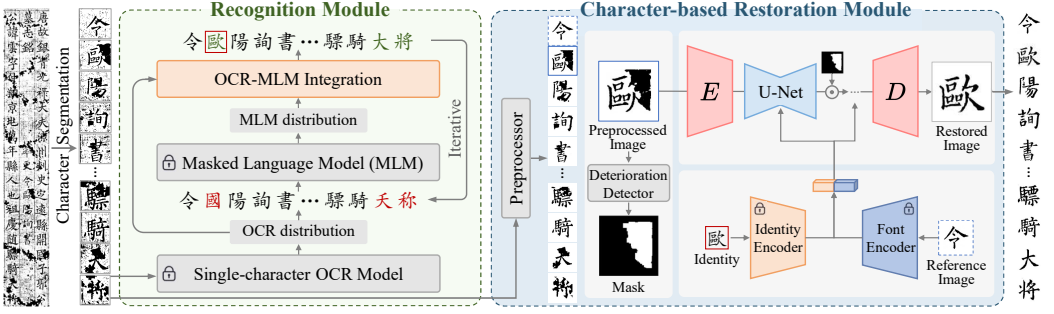


Fig. 2. Overview of our two-stage framework: a recognition module that identifies the correct identities of segmented image sequences, and a restoration module that restores the deteriorated characters based on the recognized character identities and the font guidance of any undeteriorated character image.

Recently, diffusion models [23] have achieved success in text-based image generation [41] and text-based image editing [42, 43]. They have also been applied to font generation as a conditional generation task [13, 44]. Diff-Font [13] leveraged character content, style, and strokes as guidance during denoising. FontDiffuser [44] introduced a multi-scale content aggregation block to preserve intricate strokes in complex characters, addressing challenges posed by complex structures and significant style variations. However, these methods still fail to reconstruct partially deteriorated characters and are notably slow. In comparison, we propose a multi-prior conditional diffusion model that restores partially deteriorated characters while preserving the undeteriorated parts.

3 OVERVIEW

In this work, we propose a two-stage framework to address the restoration of the rubbing image, as shown in Fig. 2. The first stage is a **recognition module** to recognize the identities of all characters in the rubbing images (including heavily deteriorated characters) (Section 4). Given a rubbing image, our method first recognizes the characters based on visual information using an OCR model (Section 4.1). However, since characters may be severely deteriorated and even damaged in a large area, we further refine character identities based on context via a masked language model (MLM) (Section 4.2). Finally, we integrate visual information based on OCR and the context based on MLM to iteratively refine character identities until convergence (Section 4.3). The second stage is a **restoration module** to recreate the appearance of all characters based on their identities and the styles of the undeteriorated characters/parts of the same rubbing (Section 5). With the character identities recognized in the first stage, we then reconstruct the appearance of the characters and recreate the rubbing image. To do so, we first apply a preprocessor to remove minor deterioration (Section 5.1). Then we propose a deterioration detector to classify pixels into deteriorated pixels and undeteriorated pixels (Section 5.2). Characters without deteriorated pixels can be used to learn the font/style of this rubbing image. Characters with deteriorated pixels will be reconstructed based on their identities and the learned font/style (Section 5.3).

4 RECOGNITION MODULE

To recognize characters in a heavily deteriorated rubbing image, single character-based OCR methods generally produce error-prone results due to partially or fully deteriorated characters. In Fig. 2, the red characters are incorrectly recognized due to heavy deterioration. To address this problem, we propose leveraging the contextual understanding ability of language models, which provide additional guidance for understanding passages and recovering visually unrecognizable

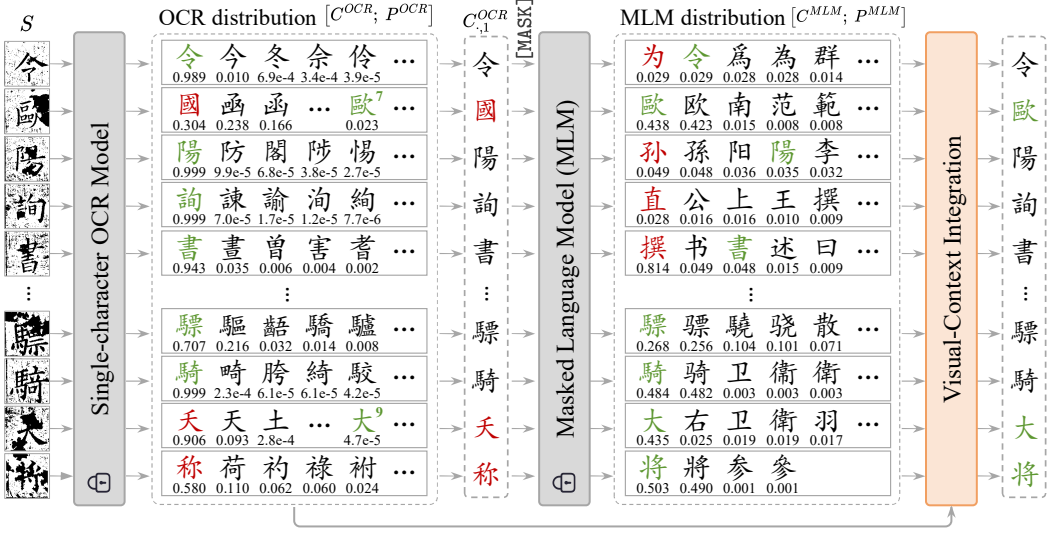


Fig. 3. One iteration of our recognition module. The number in the upper right corner of a character indicates its rank in the OCR results. For example, the “7” in the upper right corner of “歐(Ou)” means that it ranks 7th. Incorrect characters are marked in red, and corrected characters are in green. Notably, in this example, the correct result is obtained after just one iteration.

characters. However, as discussed in Section 2.3, a naive combination of language models with visual recognition may lead to error accumulation, as both models may provide degraded character probability estimations on deteriorated texts. To mitigate this problem and achieve robust character understanding from deteriorated images, we design a novel recognition module that considers text recognition from two information sources: visual cues from a finetuned OCR model and contextual cues from an MLM on Chinese classics. The recognition module revisits the candidate character probability distributions from both the OCR and the MLM sides, to conduct a fine-grained fusion of the deteriorated character probability. More importantly, we design this module to be iterative, where contextual cues and the fusion of OCR-MLM probabilities are performed several times. This iterative process gradually reduces the uncertainty of language models, improving overall recognition in long sequences of deteriorated characters. We present an overview of the recognition module on the left of Fig. 2, and a detailed illustration of the recognition process in Fig. 3.

4.1 Visual-based Character Recognition

Given a rubbing image, the first step is to recognize the characters based on visual information through an OCR model. Although sequence-based OCR methods achieve better recognition by exploiting context, we find that these methods generally cannot be used in our task for two main reasons. Firstly, sequence-based OCR methods are trained with text images in modern languages and can hardly work well on rubbing images written in ancient languages. Finetuning these models is also impossible due to the lack of human-labeled rubbing image data. Secondly, sequence-based OCR methods are not designed to handle heavily damaged image input, so their performance would be greatly affected even if the models could be finetuned. We propose using a single-character OCR model to generate the initial recognition distribution for each rubbing character image (the left part of Fig. 3), and later incorporate a language model to compose a context-aware recognition module with strong robustness to deterioration.

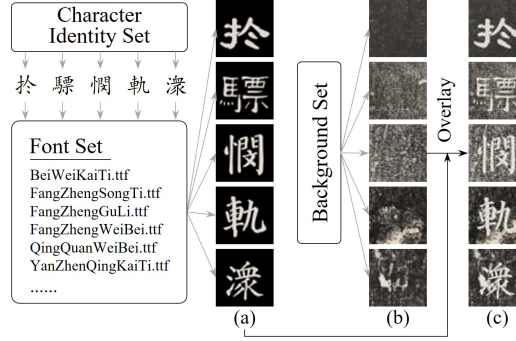


Fig. 4. Synthesis process of our recognition training dataset. (a) Computer rendered fonts; (b) Samples of extracted backgrounds; (c) Synthesis results.

4.1.1 Character Segmentation. To apply single-character OCR to the characters in a rubbing image, we first identify the bounding box of each character. Since characters in a rubbing image are typically arranged in a grid, we estimate a grid using an existing histogram projection method [45]. Experiments have shown that this approach works satisfactorily in most rubbing images. With the character grid, we extract the single-character images by sorting the grid cells in a right-to-left and top-to-bottom order, as shown in the leftmost part of Fig. 2.

4.1.2 Visual-based OCR Model. With the rubbing images segmented into character levels, we then estimate the initial likelihood of character identity using an OCR model. Existing character recognition models are generally trained on modern scene text datasets, such as the Chinese Street View Text dataset [46]. However, they produce unsatisfactory results for rubbing images. This is due to the significant visual differences between natural scene text and rubbing text in both font type and scene text context. To address this issue, we propose finetuning the state-of-the-art character OCR model, PP-OCrv3 [47], on our prepared Chinese rubbing recognition dataset.

4.1.3 Recognition Training Dataset. Currently, there is no available dataset specifically for the recognition of rubbing characters. Creating such a dataset manually is labor-intensive and requires expert knowledge to accurately identify and recover heavily deteriorated characters. To address this, we propose a reverse process to synthesize images of rubbing characters, as illustrated in Fig. 4. Specifically, we first identify the 8,000 most frequent characters from the Chinese classics corpora to form a character set and select 30 computer fonts similar to real-world rubbing fonts to form a font set. Given a character identity and a font, we render a character image with a pure black background and a foreground that approximates the rubbing character color (Fig. 4(a)). To further simulate the real deterioration in the rubbings, we manually extract character-free patches of various sizes from 15 real-world rubbing images and then randomly crop them to generate 450 patches of 48×48-resolution with different levels of deterioration to form the background (Fig. 4(b)) for rubbing synthesis. This previously rendered character image is then blended with a random background patch using inverse color code multiplication to create the final rubbing character image (Fig. 4(c)). Furthermore, we enhance the diversity of the synthesized rubbing character image by applying random rotation (skewing between -2° and 2°) and margin control, i.e., randomly shifting the center of text placement and rescaling the character to ensure the margins are between 5 and 10 pixels wide for all sides. In total, we synthesize approximately 1.3 million 48×48-resolution rubbing character images to finetune the character OCR model.

4.1.4 Training Details. We finetune the pre-trained PP-OCrv3 [47] with our synthesized rubbing character dataset, utilizing PaddlePaddle 2.4.1 [48] and training on an NVIDIA RTX 3090 GPU. During finetuning, we use the Adam optimizer [49] with β_1 set to 0.9, β_2 set to 0.999, and a learning rate of $5e^{-4}$. We adopt a batch size of 128 and train our model for 22 epochs. With the finetuned OCR model, we can estimate the OCR probability distribution for each input rubbing character image as a list of $\{text, probability\}$ pairs $[C_i^{OCR}; P_i^{OCR}]$, as shown on the left side of Fig. 3.

4.2 Text Context-based Character Prediction

Although our finetuned OCR model provides satisfactory recognition for characters with minor deterioration, it generally fails on heavily deteriorated characters. In light of this, we propose to resolve visual-level uncertainties using a bidirectional transformer-based masked language model (MLM) for its ability to robustly predict the character’s identity from its context, i.e., the adjacent characters. This robustness arises from the MLM’s deep contextual understanding, which is developed through its training schemes, allowing it to perform well even if the context contains some misrecognition, which is common in heavily deteriorated rubbing images. In our experiments, we use a state-of-the-art MLM, *roberta-classical-chinese-large-char* [50], as our foundation language model. This model was finetuned from GuwenBERT [51] using 1.7 billion characters from the classic Chinese corpora and demonstrates superior performance in understanding the context of Chinese classics compared to other existing models. It is important to note that the OCR and MLM models can be substituted with any new state-of-the-art techniques that may emerge in the future.

To recover the identity of a single character from its contexts, we mask each character in the post-OCR recognized text with a [MASK] token and then predict the likelihood distribution $[C_i^{MLM}; P_i^{MLM}]$ of the masked character based on the context, as illustrated in Fig. 3. The MLM distribution successfully predicts some rubbing characters that the OCR model incorrectly identifies, such as the second, the second last, and the last characters, where the MLM recovers the correct identities (“歐”, “大”, “将”) from the erroneous OCR outputs (“國”, “天”, “称”). However, it can also produce false predictions, as seen with the first, third, fourth, and fifth characters (“为”, “孙”, “直”, “撰”). This also shows that simply using the language model as a post-processing step for the OCR results can lead to error-prone outcomes. To overcome this, we propose integrating the OCR and MLM distributions through an iterative approach to achieve more reliable final results.

4.3 Iterative Visual-context Integration

As discussed above, neither the OCR distribution nor the MLM distribution is reliable enough on its own to provide the correct character identification results. The OCR-derived distribution may be heavily influenced by rubbing image deterioration, and the MLM-derived distribution could be skewed by incorrect context estimations made during the OCR phase. Consequently, when multiple consecutive characters are deteriorated and thus unreadable to the OCR, the MLM may struggle to accurately predict the context, since it is trained to recover only a single masked character or word at a time. To cope with the uncertainties and reinforce joint recognition of the OCR and the MLM, we design a visual-context integration mechanism by integrating the visual information provided by the OCR and the contextual information provided by the MLM in an iterative manner. Our design of the integration mechanism is supported by three observations: 1) the visual information is mostly reliable when a rubbing character has only minor deterioration; 2) contextual information is mostly reliable when a rubbing character is heavily deteriorated but its neighboring characters are correctly identified; and 3) contextual information can be propagated bidirectionally via multiple iterations.

In a round of visual-context integration, given an image sequence of rubbing characters S , let S_i denote the i -th character in the sequence. We represent the character distribution of a character image S_i as a priority list of potential characters C_i and their corresponding probabilities P_i , with the highest probability character at the top. The j -th character in this priority list and its corresponding probability are represented as $C_{i,j}$ and $P_{i,j}$, respectively. We obtain two sets of character distributions for each character image S_i : the OCR distribution $[C_i^{OCR}; P_i^{OCR}]$ and the MLM distribution $[C_i^{MLM}; P_i^{MLM}]$. We then integrate the OCR and MLM distributions by applying the following **priority rules in sequence**:

- **Rule 1:** $C_{i,1}^{MLM} = C_{i,1}^{OCR} \Rightarrow C_{i,1}^{MLM}/C_{i,1}^{OCR}$
When the top-1 characters from the OCR and MLM distributions match, this character is adopted. This usually occurs when both the character and its neighbors have minor deterioration, such as the fourth last character (“驃”) in Fig. 3.
- **Rule 2:** $P_{i,1}^{OCR} > 0.95 \Rightarrow C_{i,1}^{OCR}$
When the top-1 characters from the OCR and MLM distributions do not match, but the top-1 OCR character is highly reliable (> 0.95 likelihood), it is adopted. This usually occurs when a character is not severely deteriorated but its neighbors are, such as the third character (“陽”) in Fig. 3.
- **Rule 3:** $C_{i,j}^{MLM} = C_{i,k}^{OCR} (j \geq 2 \vee k \geq 2) \Rightarrow \min_j C_{i,j}^{MLM} = C_{i,k}^{OCR} (j \geq 2 \vee k \geq 2)$
When the top-1 characters from the OCR and MLM do not match and the OCR is not confident in its recognition probabilities, we search for overlapping characters in both distributions. The character with the highest likelihood in the MLM distribution is adopted. This typically happens when the target character cannot be recovered by OCR due to heavy deterioration, but its neighbors are less deteriorated or correctly recovered by OCR methods, such as the second and second last characters (“歐” and “大”) in Fig. 3.
- **Rule 4:** $P_{i,1}^{MLM} > 0.5 \Rightarrow C_{i,1}^{MLM}$
When there is no overlap in potential characters between the OCR and MLM distributions, but the top-1 MLM character has a likelihood over 0.5, it is chosen. This typically occurs when a character and its neighbors are deteriorated but contextual information is successfully propagated from undeteriorated parts, such as the last character (“將”) in Fig. 3.
- **Rule 5:** *OTHERWISE* $\Rightarrow C_{i,1}^{OCR}$
If none of the above rules apply, the top-1 OCR character is adopted, since visual information is generally more reliable when neither source is confident.

While the visual-context integration step works well when either visual or contextual information is reliable, it struggles with long sequences of heavily deteriorated characters. To address this, we propose an *iterative* approach to propagate correct contextual information from reliable characters to deteriorated ones step by step. The bidirectional nature of the masked language model allows contextual information to be propagated in both directions, increasing the probability of recovering a deteriorated character if reliable contextual information exists in any one direction. We illustrate this iterative approach in Fig. 5, where errors in the OCR prediction can be fixed with two complete iterations. Specifically, each iteration in our approach contains two steps. First, we perform an MLM estimation step, re-estimating the MLM-based probability as the contexts (i.e., top-1 candidates) are updated by the visual-context integration step from the previous iteration. In the second step, we perform the visual-context integration again using the initially obtained OCR probability distribution along with the updated MLM probability distribution. This process iterates until convergence, meaning that no further updates in the output top-1 candidates occur.

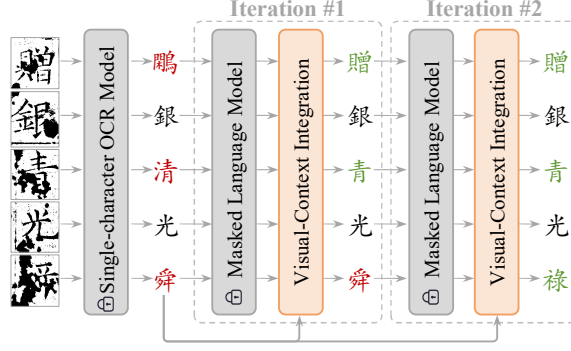


Fig. 5. Iterative approach to gradually improve the recognition accuracy. Incorrect characters are marked in red, and corrected characters are in green. The iteration continues until no further updates are made.

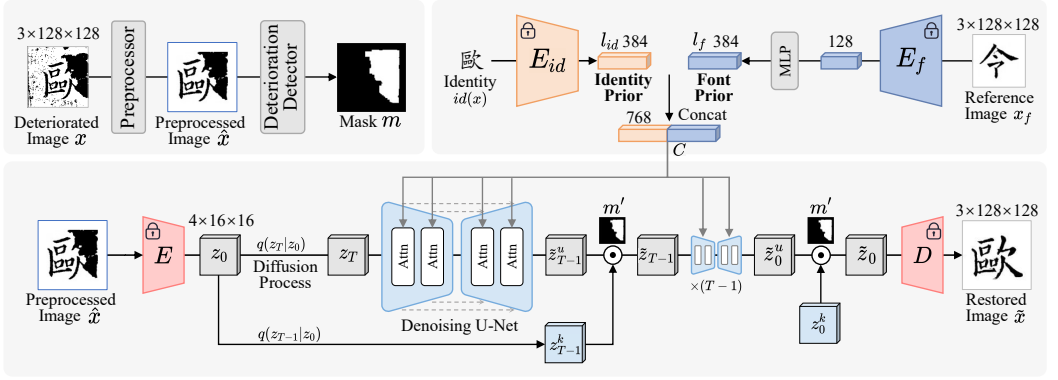


Fig. 6. Overview of our restoration module. The reference image x_f is randomly selected from the undeteriorated character images in the rubbing; m' is the result of downsampling the mask m by a factor of 8.

5 RESTORATION MODULE

Our second stage of the restoration module restores clean rubbing character images from deteriorated ones, based on the recovered character identity from the previous stage, along with the visuals of the undeteriorated parts and the fonts of the same rubbing. The key challenges of this module are twofold. Firstly, the font of each rubbing image may differ, requiring an understanding of arbitrary rubbing fonts. Secondly, the reconstructed portion of the partially deteriorated rubbing character images must be visually consistent with the undeteriorated parts. To address these challenges, we first remove small-scale deterioration using a restoration preprocessor. Next, we detect heavily deteriorated pixels that need restoration using a deterioration detector. Finally, we employ a tailored latent diffusion model with multiple priors from the identity of the rubbing character and the font of other undeteriorated characters to restore the deteriorated parts of the character image with overall visual consistency. The complete restoration module pipeline is illustrated in Fig. 6.

5.1 Restoration Preprocessor

We first propose applying a preprocessor to remove minor deterioration in the rubbing character image x to obtain the denoised image $\hat{x}$ as the initial step of the restoration module. These minor

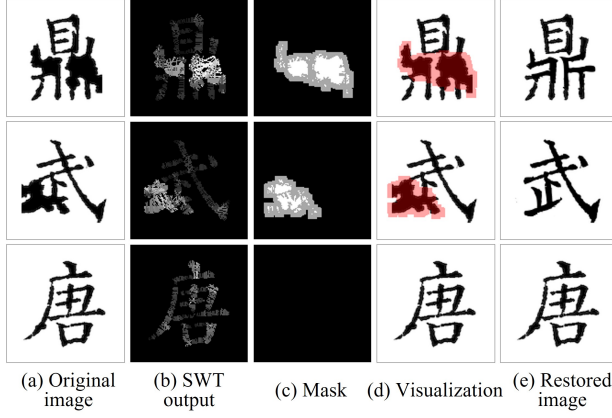


Fig. 7. Visualization of the deterioration detection process. The gray pixels in (c) are the result of dilation, and the red area in (d) corresponds to the detected deterioration mask in (c).

deteriorations may not reduce the readability of the text, but may cause visual noise such as scratches and cracks. To facilitate the next step of character restoration with diffusion models, where we require the text to have only blob-like deteriorations (as in Fig. 7) to lower the learning cost, this preprocessing step is essential to ensure that minor scratches and noise are removed.

To handle similar levels of minor deterioration, various denoising methods [2, 17, 27] have been proposed. Among them, the GAN-based calligraphic image denoising method CIDG [2] slightly outperforms others in removing minor noise from rubbing images. As a result, we train a customized CIDG model with a newly synthesized rubbing denoising dataset. As shown in Fig. 6, the resulting denoiser produces a noise-free image in which minor deterioration is removed. For more details on dataset preparation and the training process, please refer to Appendix Section A.1.

5.2 Deterioration Detector

In addition to minor scratches and defects, the weathering process can cause erosion, tearing material off the rubbings and creating relatively more severe blob-shaped visual artifacts, as shown in Fig. 1 and Fig. 7. These intermediate to severe deteriorations are a primary cause of illegibility for both human readers and automated OCR approaches. To resolve deterioration at this level, it is necessary to locate and restore the deteriorated areas with the remaining pixels untouched. Therefore, we propose a deterioration detector to classify the deteriorated pixels in the preprocessed denoised image $\hat{x}$ and represent this classification as a binary deterioration mask m .

We build this classifier on the stroke width transform method [52], which determines the stroke width at each character pixel in an image. Fig. 7(b) visualizes the stroke width values calculated by this method, where brighter colors indicate larger stroke widths. In cases of intermediate to severe blob-like deterioration, the stroke widths of deteriorated pixels are typically much larger than those of undeteriorated pixels. Therefore, for each rubbing image, we calculate the stroke width distribution of all foreground pixels and then identify the outlier pixels. This approach is based on the assumption that characters within the same rubbing image are typically written in similar styles and, consequently, have similar stroke widths. Therefore, even if some characters exhibit complete or significant damage to their strokes, we can still estimate the overall stroke width of the rubbing image by referencing the less deteriorated characters. Furthermore, since different rubbing images may have varying fonts and font sizes, we set an adaptive detection threshold t_w to be the

sum of the mean μ_w and the standard deviation σ_w of the stroke width values of the foreground pixels, i.e., $t_w = \mu_w + \sigma_w$. Foreground pixels with stroke width values greater than t_w are identified as deteriorated. We further dilate these pixels with a kernel size of 10 to connect isolated detection regions and form the deteriorated regions shaped like blobs. Fig. 7(c) shows the detection, where pure white pixels represent deteriorated pixels with stroke widths greater than t_w , and gray pixels are dilated pixels. Based on the visualized examples, no false negatives are observed when applying our threshold design and dilation workflow.

5.3 Character Restoration with Visual and Contextual Priors

To restore deteriorated rubbing images, we first learn a latent diffusion model θ that generates clean rubbing character images $\tilde{x}$ with the identity and font of the character specified as model conditions C . During character restoration, this model guides the transformation of any deteriorated image $\hat{x}$ into the distribution of $\tilde{x}$, resulting in a clean rubbing character image.

We illustrate our proposed character restoration process in Fig. 6. The key insight is that, in addition to the condition C , we use the undeteriorated pixels $(1 - m) \odot \hat{x}$ as an additional shape constraint to guide the denoising diffusion of the deteriorated pixels $m \odot \hat{x}$ for a globally consistent structure of the restored characters. Here, $\odot$ represents pixel-wise multiplication and m is the deterioration mask detected in the previous step. Practically, we update the latents that have deteriorated according to the U-Net output $\epsilon_\theta(z_t, t, C)$ during the denoising diffusion process and update the undeteriorated latents according to the forward noise schedule $q(z_{t-1}|z_0)$, as the clean values of these latents are already given by the undeteriorated pixels of the input image. This design maintains visual consistency for the deteriorated latents (and thus, pixels) and removes the need for paired supervision, as the denoising diffusion model θ automatically drives the distribution of the deteriorated pixels towards that of the undeteriorated ones, similar to the concepts demonstrated in SDEdit [42].

5.3.1 Model Architecture. We adopt the methodology of latent diffusion models [53] and train a U-Net-based network θ with the cross-attention mechanism [54]. Specifically, given a character image $I \in \mathbb{R}^{H \times W \times 3}$, the encoder E transforms it into a latent representation $z_0 = E(I) \in \mathbb{R}^{h \times w \times 4}$, where H and W represent the height and width of I , and h and w represent the height and width of z_0 . We denote the downsampling factor by $f = H/h = W/w$, setting $f = 8$ to balance quality and resource consumption.

To inject the identity and font priors into the diffusion pipeline, we introduce two components: an identity encoder (denoted as E_{id}) and a font encoder (denoted as E_f). The identity encoder utilizes an identity embedding layer to provide the identity prior. We did not choose pre-trained BERT embeddings as we are focusing on single-character identity modeling. Specifically, E_{id} reads the character identity $id(x)$ recovered from our recognition module and maps it to a 384-dimensional word embedding, denoted l_{id} . The font encoder, borrowed from DG-Font [11], is based on VGG-11 [55]. This encoder takes a randomly selected reference character image x_f and produces a 128-dimensional font representation. We further transform this latent into 384-dimensional with a two-layer MLP to balance the two different priors. These two latent vectors are then concatenated to form a latent C as a condition, represented as $C = \text{concat}(E_{id}(id(x)), E_f(x_f))$. Later in the denoising diffusion steps, we map the condition C to the intermediate layers of the U-Net through a cross-attention layer. In this setup, we learn the *key* and *value* from condition C and learn the *query* from $\phi(z_t)$, where z_t is a noisy version of z_0 obtained through a fixed forward diffusion process ($t \in [1, ..., T]$) and $\phi(z_t)$ denotes an intermediate representation of the U-Net (see Appendix

Section B for more detailed model architecture). Based on this, the objective can be formulated as:

$$L := \mathbb{E}_{z, C, \epsilon \sim \mathcal{N}(0,1), t} \left[\|\epsilon - \epsilon_\theta(z_t, t, C)\|_2^2 \right], \quad (1)$$

where ϵ is the added known noise. After T denoising steps, we obtain $\tilde{z}_0$, which is then decoded by the decoder D back into pixel space, i.e., $\tilde{x} = D(\tilde{z}_0)$. During training, the identity encoder E_{id} and the font encoder E_f are kept frozen.

5.3.2 Restoration Training Dataset. During our experiments, we find that the generation of rubbing images is much more difficult than the recognition, as generalizing to more diverse fonts and writing styles is critical to reconstruct and restore the rubbings of different fonts and deterioration levels. As a result, instead of using the recognition dataset produced in the previous stage of character recognition, we created a much larger font dataset with computer fonts from the internet¹ and ensured that these fonts can generate characters that can be recognized by our recognition module. We collect 380 computer fonts to build a restoration training dataset. Our vocabulary aligns with our recognition training dataset (Section 4.1.3), covering 8,000 character identities with the highest frequency among Chinese classics. We render the characters in the center of a 128×128 canvas with a fixed font size without further augmentation. The resulting dataset includes approximately 2.5 million character images.

5.3.3 Training Details of Diffusion Model. For the training of our restoration module, we observe a considerable gap in utilizing existing latent diffusion VAEs for rubbing reconstruction, since they were typically trained on the OpenImage natural image dataset [56], and therefore struggle to reconstruct and generate rubbing characters (as discussed in Appendix Section C.4). To address this, we further finetune the kl-f8 VAE² in [53] using our restoration training dataset. The finetuning is implemented using PyTorch and runs for two epochs on two NVIDIA RTX 3090 GPUs with a learning rate of $4.5e^{-6}$ and a batch size of 26. Subsequently, we train a latent diffusion model from scratch with the previously finetuned VAE encoder and decoder. We use the AdamW [57] optimizer with a learning rate set to $1e^{-4}$. For each character image $\tilde{x}$ to be generated, we use its ground truth identity $id(x)$ to train the injection of identity prior. We also sample another character image with the same font as the input as the reference image x_f , providing the font prior. Following the classifier-free guidance method [58], we randomly drop out priors $id(x)$ and x_f with a probability of 0.1 to improve the model’s robustness under partially provided conditions. The batch size is set to 64 and the model is trained with 600,000 iterations.

5.3.4 Finetuning on Target Rubbing Style. Our trained diffusion model is expected to restore any deteriorated rubbing characters when the conditions are given correctly. However, our experiments show that this base model may hallucinate unnecessary details or generate inconsistent stroke shapes during rubbing character generation (Fig. 8 (b)), when a real rubbing image is given as font guidance. We attribute this issue to the distribution mismatch between the computer-generated fonts in the training dataset and the real rubbing images. When referenced by a real rubbing image, the base model may not be able to link this reference to any font it has seen during training, resulting in a degradation of generation quality. To provide tailored generation and reconstruction quality for the target rubbing font, we may perform additional rubbing-specific finetuning. To do so, we randomly select at least 10 undeteriorated characters from the same volume of rubbings crafted by the same artist and label the identity of these characters either manually or using our previously introduced recognition module. With these few-shot examples, finetuning the model with around 4,000 iterations, taking less than 1 hour on an NVIDIA RTX 3090, is sufficient to

¹<https://www.foundertype.com/>

²<https://ommer-lab.com/files/latent-diffusion/kl-f8.zip>



Fig. 8. Investigation of finetuning on the diffusion-based character restoration model. Each character in (b) and (c) is generated from the diffusion model using the ground truth identity and style references from (a). Better results should exhibit stroke styles and font topology similar to (a).

achieve a noticeable improvement in visual quality, as shown in Fig. 8 (c). Given the challenges of manual restoration, we believe that this optional finetuning process is cost-effective to achieve fully automated rubbing restoration with improved visual quality and font similarity to the reference.

5.3.5 Details of Rubbing Restoration Pipeline. As illustrated in Fig. 6, we input the preprocessed rubbing image $\hat{x}$ into the latent space to produce z_0 . To restore pixels that have deteriorated, we initially add sufficient noise to the entire image to obtain pure noise z_T , where $T = 25$ denotes the number of DDIM sampling steps used at inference, calculated as:

$$z_T \sim \mathcal{N}(\sqrt{\bar{\alpha}_T}z_0, (1 - \bar{\alpha}_T)\mathbf{I}) \quad (2)$$

Subsequently, we continuously manipulate the denoising diffusion to obtain $\tilde{z}_0$ in the following procedure: at each step t ($t \in [T, \dots, 1]$), we first denoise the noisy latent vector z_t under the guidance of condition C , resulting in a less noisy unknown part, denoted as $\tilde{z}_{t-1}^u$ (Eq. 3a). Meanwhile, we employ the fixed forward diffusion process to diffuse z_0 to the current noise level, yielding the known part z_{t-1}^k (Eq. 3b). Then we blend the two latents to obtain the fused latent $\tilde{z}_{t-1}$ for the next step using the mask m' , which is downsampled from m with a factor of 8 (Eq. 3c).

$$\tilde{z}_{t-1}^u \sim \mathcal{N}(\mu_\theta(z_t, t, C), \Sigma_\theta(z_t, t, C)) \quad (3a)$$

$$z_{t-1}^k \sim \mathcal{N}(\sqrt{\bar{\alpha}_t}z_0, (1 - \bar{\alpha}_t)\mathbf{I}) \quad (3b)$$

$$\tilde{z}_{t-1} = m' \odot \tilde{z}_{t-1}^u + (1 - m') \odot z_{t-1}^k \quad (3c)$$

Repeating Eq. 3 for T steps, we obtain the final $\tilde{z}_0$, which is then decoded into the pixel space to result in the final restored image $\hat{x}$. For more details on the sampling process, please refer to the Appendix Section A.2.

6 EXPERIMENTS

To demonstrate the effectiveness of our method in restoring deteriorated real-world rubbing images, we first present two heavily deteriorated examples and our restored results in Fig. 9. The examples contain partially and even fully deteriorated characters. Our method not only correctly restores the text but also produces visually consistent fonts for restored characters. In addition, we conducted in-depth evaluations to compare our method with existing methods both visually and quantitatively. Note that no existing solution handles the recognition and restoration of deteriorated character images in a complete pipeline, as we do. Therefore, we roughly classify existing methods into four categories: denoising [2, 17, 27], inpainting [5, 6], character recognition [47, 59], and font



Fig. 9. Restoration results of real-world rubbing images. Credits: (1) *Guo Yun mu zhi*. Tang Dynasty, Chinese; (2) *Lanting ji*. Tang Dynasty, Chinese. Source: [Chinese Rubbings Collection - CURIOSity Digital Collections](https://curiosity.lib.harvard.edu/chinese-rubbings-collection).

generation [11–13], and compare the corresponding parts of our pipeline with each. Finally, we evaluate our systematic designs with ablation studies in Appendix Section C.

6.1 Test Dataset

To ensure a comprehensive and fair comparison, we construct two test datasets: a recognition test dataset and a restoration test dataset.

6.1.1 Recognition Test Dataset. We prepare 100 complete rubbing images from the high-resolution scanned images in the Harvard Library collection³. These rubbings, primarily from the Wei and Tang dynasties of China, are 1,000 to 2,000 years old. Using the character segmentation method detailed in Section 4.1.1, we segment these rubbing images into 30,112 single-character images with varying levels of deterioration. We manually annotate the ground truth identities of these characters to form our recognition test dataset.

6.1.2 Restoration Test Dataset. We created a restoration test dataset consisting of pairs of deteriorated rubbing character images and their corresponding ground truth. Ideally, we can obtain the ground truth by human labeling from the recognition test dataset. However, manually restoring severely deteriorated rubbing character images proves to be extremely difficult and unreliable.

³<https://curiosity.lib.harvard.edu/chinese-rubbings-collection>

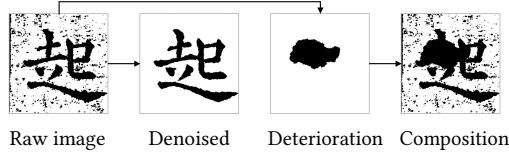


Fig. 10. Synthesis process of our restoration test dataset.

Therefore, we propose a reverse-engineering dataset that synthesizes deterioration from rubbing images with minor deterioration, as shown in Fig. 10. Specifically, we selected 592 minor deteriorated rubbing character images from the collected 100 high-resolution scanned rubbing images. For each character, we manually denoise the image to obtain the ground truth and then place random severe deterioration extracted from different characters using our deterioration detector (Section 5.2), with random rotation, translation, and scaling. All image pairs have a resolution of 128×128 .

6.2 Comparisons with Denoising and Inpainting Methods

Image denoising aims to remove noise from images, where the deterioration in rubbing images can be seen as noise. Here we compare our method with two state-of-the-art denoising methods tailored for rubbing images, CIDG [2] and CharFormer [27]. We also compare with a general image denoising approach, Uformer [17]. Additionally, inpainting methods fall within our scope as they are capable of synthesizing new image content in the masked deteriorated areas. As no inpainting methods have been proposed for rubbing images, we evaluate our method against two state-of-the-art image inpainting techniques: transformer-based MAT [5] and diffusion-based RePaint [6]. For fair comparison, we train all our competitor models from scratch with the same restoration training dataset. Since inpainting methods assume a noise-free input and aim to restore contents only in the masked region, we use the noise-free rubbing character image computed by our restoration preprocessor as the model input. Additionally, we use the detected deterioration mask of our deterioration detector as the user-specified inpainting mask.

6.2.1 Visual Comparison. We illustrate the visual comparison in Fig. 11. We use a dashed line to separate denoising model results from inpainting model results. From the results, we observe that all denoising models effectively remove minor deterioration. However, for severe deterioration, they either retain large-scale blotches or transform some blotches into empty white pixels without strokes, failing to achieve our goal of filling in topology and appearance with contiguous strokes. This behavior is expected for denoising models as they focus solely on noise removal and do not learn to generate strokes covered by large-scale blotches. On the other hand, inpainting methods can reconstruct font-consistent strokes for severely deteriorated characters; but their results often deviate significantly in stroke topology from the ground truth, making the recovered characters unrecognizable. This discrepancy arises from the lack of consideration of character identity during the inpainting process. As in the sixth column of Fig. 11, only our method correctly restores the right part (“攴”) of the character (“攴”).

6.2.2 Quantitative Comparison. In the quantitative evaluation, we measure the similarity between the restored images and the ground truth using pixel-based metrics, including L1 distance and RMSE, as well as the perceptual-based similarity metric LPIPS [60], which aligns closely with human perception. Additionally, we calculate the Chamfer Distance (CD) [61] to reflect the shape



Fig. 11. Visual comparison with the state-of-the-art denoising and inpainting methods. Our model is finetuned with characters from the same rubbing.

Table 1. Quantitative comparison with the state-of-the-art denoising and inpainting methods. Bold for optimal values and underline for suboptimal.

	L1 ↓	RMSE ↓	LPIPS ↓	CD ↓	OCR% ↑
Deteriorated Image	0.2180	0.4187	0.5894	8.33	34.37
Uformer [17]	0.1410	0.3594	0.2430	6.03	37.11
CharFormer [27]	0.0983	0.2785	0.1985	3.60	42.86
CIDG [2]	0.0873	0.2715	0.2045	4.07	39.40
RePaint [6]	0.0578	0.2208	0.1469	2.58	43.69
MAT [5]	0.0546	0.2028	0.1387	2.38	46.08
Ours (w/o finetuning)	<u>0.0538</u>	<u>0.2017</u>	<u>0.1185</u>	<u>2.10</u>	<u>91.76</u>
Ours (w/ finetuning)	0.0467	0.1868	0.0998	1.82	92.73

distance between the restoration and the ground truth, formulated as:

$$d_{CD}(S_1, S_2) = \frac{1}{|S_1|} \sum_{x \in S_1} \min_{y \in S_2} \|x - y\|_2^2 + \frac{1}{|S_2|} \sum_{y \in S_2} \min_{x \in S_1} \|y - x\|_2^2, \quad (4)$$

where S_1 represents the set of non-zero pixels in the restored character image, and S_2 represents the set in the ground truth ($S_1, S_2 \subseteq \mathbb{R}^2$). Additionally, we measure recognition accuracy using our trained character-only OCR model (the finetuned PP-OCRv3 model introduced in Section 4.1) to reflect how our restoration model reduces uncertainty in character recognition.

Table 2. Evaluation of recognition accuracy (%) on rubbings.

Methods	Minor Deterioration	Severe Deterioration	Overall
EasyOCR [59]	61.1	46.0	52.1
PP-OCrv3 [47]	72.8	57.6	63.8
Finetuned PP-OCrv3	87.4	74.8	79.9
Ours	93.7	85.5	88.8

Table 1 presents the quantitative results. For denoising methods, although the removal of minor deterioration brings some improvements, particularly rubbing-tailored methods [2, 27], these methods fail to handle severe deteriorations. Consequently, their results show a significant gap compared to ours. Compared with inpainting methods, we achieve much better performance in LPIPS, Chamfer Distance, and OCR accuracy. This is mainly because inpainting methods lack explicit character identity guidance, making it difficult to generate the correct stroke topology. We would highlight that the improvements in L1 and RMSE are marginal because our stroke restoration cannot be pixel-aligned to the ground truth due to the loss of information. We observe slight deviations in the strokes of the generated characters compared to the ground truth, but the overall font remains consistent (as shown in the last two rows of Fig. 11, especially the character “左”). This is acceptable, as real-life restoration should produce visually natural, harmonious, and sensible fonts when multiple solutions are possible. Moreover, we also demonstrate additional performance improvements with rubbing-specific finetuning. We perform separate finetuning for each rubbing and calculate their weighted averages as the final metrics. Finetuning helps direct the restored font closer to the target rubbing font, making the performance improvement evident and intuitive.

6.3 Comparisons with Character Recognition Methods

We compare our recognition module with state-of-the-art Chinese character recognition methods, including EasyOCR [59] and PP-OCrv3 [47]. Since the pre-trained PP-OCrv3 is trained on modern scene text datasets and does not perform well on rubbing recognition, we also compare with a PP-OCrv3 model finetuned using our recognition training dataset. We also include a visual comparison with the state-of-the-art vision language model PaddleOCR-VL [37] in Fig. 12. All experiments were conducted on our recognition test dataset.

6.3.1 Visual Comparison. Fig. 12 presents examples and recognition results from the finetuned PP-OCrv3, PaddleOCR-VL, and our recognition module. We observe that relying solely on visual representation often leads to recognition errors, even with very minor deterioration. Visual-only OCR models are error-prone in distinguishing visually similar characters. For example, PP-OCrv3 incorrectly recognized the character “洛” as “浴”. Vision language models are also prone to errors when working with rubbing images due to the significant domain gap between natural text and rubbing characters, particularly in areas where deterioration is present. For example, PaddleOCR-VL incorrectly recognized “鼎” as “吊”. In comparison, our method successfully recognized all these characters by integrating visual and contextual information.

6.3.2 Quantitative Comparison. We evaluate the recognition accuracy of our method and the competitors on our recognition test dataset. We do not include the quantitative comparison result

⁴Credits (from left to right): (1) *Yuan Shouan mu zhi*. Northern Wei Dynasty, Chinese; (2) *Huan Zhongguang mu zhi*. Tang Dynasty, Chinese; (3) *Lanting ji*. Tang Dynasty, Chinese; (4)-(8) *Guo Yun mu zhi*. Tang Dynasty, Chinese. Source: [Chinese Rubbings Collection - CURIOSity Digital Collections](#).

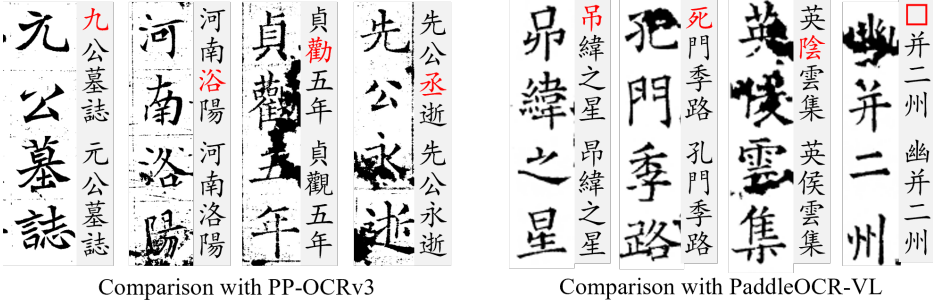


Fig. 12. Visual comparisons with finetuned PP-OCRv3 and PaddleOCR-VL on recognition accuracy. Predictions are placed on the right side of the rubbing images, with incorrect predictions highlighted in red. In each example, the upper group of characters is the prediction of the competing method (finetuned PP-OCRv3 on the left, PaddleOCR-VL on the right), and the lower group is ours.⁴

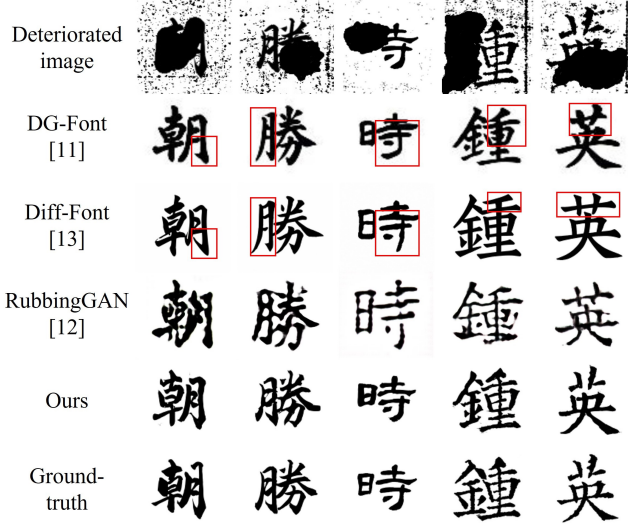


Fig. 13. Visual comparison with the state-of-the-art font generation methods.

with PaddleOCR-VL, as its recognition outputs fail to yield text sequences following the proper reading order. Characters from distinct lines are frequently interleaved. Directly computing the recognition accuracy based on such outputs produces extremely low metrics ($< 20\%$). For this reason, PaddleOCR-VL is excluded from our quantitative evaluation. As shown in Table 2, our method significantly outperforms other existing methods in recognition accuracy.

6.4 Comparisons with Font Generation Methods

Lastly, we compare our restoration module with font generation methods, including the rubbing-tailored font generation method RubbingGAN [12], as well as generic text-conditioned font generation methods DG-Font [11] and Diff-Font [13]. For fair comparisons, we train DG-Font and Diff-Font from scratch using our restoration training dataset. For RubbingGAN, we train separate models for different fonts as required, matching its data needs.

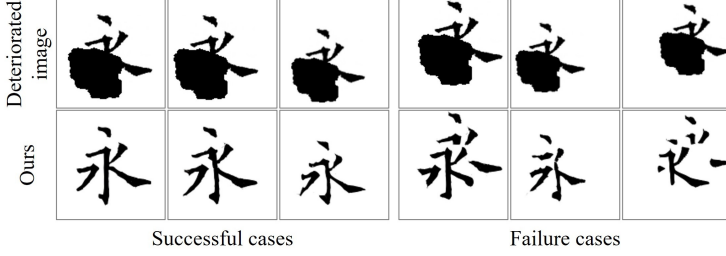


Fig. 14. Restoration of deteriorated images at different positions and sizes.

Table 3. Ablation study on the amount of restoration training data. Bold for optimal values.

	L1 ↓	RMSE ↓	LPIS ↓	CD ↓	OCR% ↑
380 fonts (all training data)	0.0467	0.1868	0.0998	1.82	92.73
190 fonts (1/2 training data)	0.0490	0.1882	0.1062	1.98	92.23
95 fonts (1/4 training data)	0.0515	0.1974	0.1142	2.12	90.84

For evaluation, we find that quantitative comparisons are challenging due to the inherent limitations in these methods. DG-Font supports only a resolution of 80×80 and cannot scale to 128×128 . RubbingGAN requires over 200 character images per rubbing, which many rubbings cannot provide. Therefore, we focus on visual comparisons, as presented in Fig. 13. We can observe that although DG-Font and Diff-Font can generate characters with consistent style and stroke topology, they fail to preserve the undeteriorated parts of partially deteriorated characters. RubbingGAN struggles with distorted and inaccurate strokes due to the instability of adversarial training and the limited character samples in one rubbing image. In contrast, our method leverages the deteriorated image for correction throughout the restoration process, allowing better preservation of undeteriorated parts. Additionally, for font generation methods, including RubbingGAN, the identity corresponding to the deteriorated image must be specified manually. In contrast, our method can automatically recognize the identity of the characters, thanks to our recognition module.

6.5 Discussions

6.5.1 Ablation on Restoration Training Dataset. We have also conducted an analysis of the amount of restoration training data, as shown in Table 3. It can be observed that increasing the amount of training data leads to improved performance, while the gains become less significant once more than 200 fonts are used for training.

6.5.2 Limitations. Our diffusion model performs best when the character images are roughly uniform in size and centered, since the training data follow this pattern (Fig. 14). Deviations from this pattern can hinder the model’s ability to effectively integrate deteriorated and undeteriorated parts. For example, in the fourth column of Fig. 14, a deteriorated image positioned too far to the upper right results in misaligned generated strokes. Although most single character images from our segmentation algorithm in Section 4.1.1 meet these criteria, some deteriorated images with significant offsets require simple corrections.

6.5.3 Extension to Other Languages. While this work focuses on ancient Chinese character rubbings, it can be extended to any language with available language and OCR models. Our method can be adapted to Greek and Roman rubbings with a modified character segmentation strategy. For

glyph-based languages without formal language models, such as Egyptian hieroglyphs, we could temporarily translate them into BERT-compatible languages like English for contextual recognition, refining OCR results. This direction will be explored in future work.

7 CONCLUSION

In this research, we introduce a novel two-stage framework for rubbing image restoration with minimal manual intervention. Our method enhances character recognition accuracy by integrating visual and contextual information and uses a multi-prior conditional diffusion model to restore deteriorated characters while preserving undeteriorated parts. Extensive experiments have demonstrated the effectiveness of our approach compared to existing methods, resulting in significant improvements in both recognition accuracy and visual quality. We plan to extend our proposed model to handle character images with diverse positional offsets and sizes, to support multiple languages and larger character sets, to develop sequence-based or entire rubbing image-based restoration algorithms, and to explore stroke segmentation and animation techniques for calligraphy rubbings to achieve even higher restoration fidelity.

Acknowledgments

This work was supported by the Research Grants Council of the Hong Kong SAR, China (Project No. UGC/FDS11/E03/22), the National Natural Science Foundation of China (No. 62273241), Natural Science Foundation of Guangdong Province, China (No. 2024A1515011946), the Shenzhen Research Foundation for Basic Research, China (No. JCYJ2025060418190054), and the National Science and Technology Council, Taiwan under Grant 114-2221-E-006-114-MY3.

References

- [1] Zhi-Kai Huang, Zhen-Ning Wang, Jun-Mei Xi, and Ling-Ying Hou. 2019. Chinese rubbing image binarization based on deep learning for image denoising. In *International Conference on Control and Computer Vision*. 46–50.
- [2] Jiulong Zhang, Mingtao Guo, and Jianping Fan. 2020. A novel generative adversarial net for calligraphic tablet images denoising. *Multimedia Tools and Applications* 79 (2020), 119–140.
- [3] Daqian Shi, Xiaolei Diao, Hao Tang, Xiaomin Li, Hao Xing, and Hao Xu. 2022. RCRN: Real-world Character Image Restoration Network via Skeleton Extraction. In *ACM International Conference on Multimedia*. 1177–1185.
- [4] Roman Suvorov, Elizaveta Logacheva, Anton Mashikhin, Anastasia Remizova, Arsenii Ashukha, et al. 2022. Resolution-robust large mask inpainting with Fourier convolutions. In *IEEE Conference on Computer Vision and Pattern Recognition*. 2149–2159.
- [5] Wenbo Li, Zhe Lin, Kun Zhou, Lu Qi, Yi Wang, and Jiaya Jia. 2022. MAT: Mask-aware transformer for large hole image inpainting. In *IEEE Conference on Computer Vision and Pattern Recognition*. 10758–10768.
- [6] Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, and Radu Timofte. 2022. RePaint: Inpainting using denoising diffusion probabilistic models. In *IEEE Conference on Computer Vision and Pattern Recognition*. 11461–11471.
- [7] Baoguang Shi, Xiang Bai, and Cong Yao. 2016. An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39, 11 (2016), 2298–2304.
- [8] Yongkun Du, Zhineng Chen, Caiyan Jia, Xiaoting Yin, Tianlun Zheng, Chenxia Li, Yuning Du, and Yu-Gang Jiang. 2022. SVTR: Scene text recognition with a single visual model. *arXiv preprint arXiv:2205.00159* (2022).
- [9] Thi Tuyet Hai Nguyen, Adam Jatowt, Nhu-Van Nguyen, Mickael Coustaty, and Antoine Doucet. 2020. Neural machine translation with BERT for post-OCR error detection and correction. In *ACM/IEEE Joint Conference on Digital Libraries*. 333–336.
- [10] Elizabeth Soper, Stanley Fujimoto, and Yen-Yun Yu. 2021. BART for post-correction of OCR newspaper text. In *Workshop on Noisy User-generated Text*. 284–290.
- [11] Yangchen Xie, Xinyuan Chen, Li Sun, and Yue Lu. 2021. DG-Font: Deformable generative networks for unsupervised font generation. In *IEEE Conference on Computer Vision and Pattern Recognition*. 5130–5140.
- [12] Gongbo Sun, Zijie Zheng, and Ming Zhang. 2022. End-to-End Rubbing Restoration Using Generative Adversarial Networks. *arXiv preprint arXiv:2205.03743* (2022).

- [13] Haibin He, Xinyuan Chen, Chaoyue Wang, Juhua Liu, Bo Du, Dacheng Tao, and Yu Qiao. 2024. Diff-Font: Diffusion model for robust one-shot font generation. *International Journal of Computer Vision* 132, 11 (2024), 5372–5386.
- [14] Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. 2017. Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising. *IEEE Transactions on Image Processing* 26, 7 (2017), 3142–3155.
- [15] Yulun Zhang, Yapeng Tian, Yu Kong, Bineng Zhong, and Yun Fu. 2020. Residual dense network for image restoration. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 43, 7 (2020), 2480–2495.
- [16] Shen Cheng, Yuzhi Wang, Haibin Huang, Donghao Liu, Haoqiang Fan, and Shuaicheng Liu. 2021. NBNNet: Noise basis learning for image denoising with subspace projection. In *IEEE Conference on Computer Vision and Pattern Recognition*. 4896–4906.
- [17] Zhendong Wang, Xiaodong Cun, Jianmin Bao, Wengang Zhou, Jianzhuang Liu, and Houqiang Li. 2022. Uformer: A general u-shaped transformer for image restoration. In *IEEE Conference on Computer Vision and Pattern Recognition*. 17683–17693.
- [18] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, and Fahad Shahbaz Khan. 2022. Restormer: Efficient transformer for high-resolution image restoration. In *IEEE Conference on Computer Vision and Pattern Recognition*. 5728–5739.
- [19] Kang Xu, Weixin Li, Xia Wang, Xiaoyan Hu, Ke Yan, Xiaojie Wang, and Xuan Dong. 2023. CUR transformer: A convolutional unbiased regional transformer for image denoising. *ACM Transactions on Multimedia Computing, Communications, and Applications* 19, 3 (2023), 1–22.
- [20] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. *Advances in Neural Information Processing Systems* 27 (2014).
- [21] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, and Trevor Darrell. 2016. Context encoders: Feature learning by inpainting. In *IEEE Conference on Computer Vision and Pattern Recognition*. 2536–2544.
- [22] Jiayu Lin and Yuan-Gen Wang. 2024. TSFormer: Tracking Structure Transformer for Image Inpainting. *ACM Transactions on Multimedia Computing, Communications, and Applications* (2024).
- [23] Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems* 33 (2020), 6840–6851.
- [24] Shen-Zheng Wang and Hsi-Jian Lee. 2001. Dual-binarization and anisotropic diffusion of Chinese characters in calligraphy documents. In *International Conference on Document Analysis and Recognition*. IEEE, 271–275.
- [25] Basilios Gatos and Ioannis Pratikakis. 2006. Adaptive degraded document image binarization. *Pattern Recognition* 39, 3 (2006), 317–327.
- [26] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. 2017. Image-to-image translation with conditional adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition*. 1125–1134.
- [27] Daqian Shi, Xiaolei Diao, Lida Shi, Hao Tang, Yang Chi, Chuntao Li, and Hao Xu. 2022. CharFormer: A glyph fusion based attentive framework for high-precision character image denoising. In *ACM International Conference on Multimedia*. 1147–1155.
- [28] Zhenhua Yang, Dezhi Peng, Yongxin Shi, Yuyi Zhang, Chongyu Liu, and Lianwen Jin. 2025. Predicting the Original Appearance of Damaged Historical Documents. In *AAAI Conference on Artificial Intelligence*. AAAI Press, 9382–9390.
- [29] Hongtao Xie, Shancheng Fang, Zheng-Jun Zha, Yating Yang, Yan Li, and Yongdong Zhang. 2019. Convolutional attention networks for scene text recognition. *ACM Transactions on Multimedia Computing, Communications, and Applications* 15, 1s (2019), 1–17.
- [30] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 4171–4186.
- [31] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In *Annual Meeting of the Association for Computational Linguistics*. 7871–7880.
- [32] Shipeng Zhu, Hui Xue, Na Nie, Chenjie Zhu, Haiyue Liu, and Pengfei Fang. 2024. Reproducing the Past: A Dataset for Benchmarking Inscription Restoration. In *ACM International Conference on Multimedia*. ACM, 7714–7723.
- [33] Deli Yu, Xuan Li, Chengquan Zhang, Tao Liu, Junyu Han, Jingtuo Liu, and Errui Ding. 2020. Towards accurate scene text recognition with semantic reasoning networks. In *IEEE Conference on Computer Vision and Pattern Recognition*. 12113–12122.
- [34] Shancheng Fang, Hongtao Xie, Yuxin Wang, Zhendong Mao, and Yongdong Zhang. 2021. Read like humans: Autonomous, bidirectional and iterative language modeling for scene text recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*. 7098–7107.
- [35] Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, et al. 2023. Qwen-VL: A Versatile Vision-Language Model for Understanding, Localization, Text Reading, and Beyond. *arXiv preprint arXiv:2308.12966* (2023).

- [36] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, et al. 2024. Qwen2-VL: Enhancing Vision-Language Model’s Perception of the World at Any Resolution. *arXiv preprint arXiv:2409.12191* (2024).
- [37] Cheng Cui, Ting Sun, Suyin Liang, Tingquan Gao, Zelun Zhang, et al. 2025. PaddleOCR-VL: Boosting Multilingual Document Parsing via a 0.9B Ultra-Compact Vision-Language Model. *arXiv:2510.14528* [cs.CV]
- [38] Cheng Cui, Ting Sun, Suyin Liang, Tingquan Gao, Zelun Zhang, et al. 2026. PaddleOCR-VL-1.5: Towards a Multi-Task 0.9B VLM for Robust In-the-Wild Document Parsing. *arXiv:2601.21957* [cs.CV]
- [39] Yuchen Tian. 2017. zi2zi: Master Chinese Calligraphy with Conditional Adversarial Networks. <https://github.com/kaonashi-tyc/zi2zi>
- [40] Chi Wang, Min Zhou, Tiezheng Ge, Yuning Jiang, Hujun Bao, and Weiwei Xu. 2023. CF-Font: Content Fusion for Few-shot Font Generation. In *IEEE Conference on Computer Vision and Pattern Recognition*. 1858–1867.
- [41] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. 2023. Adding conditional control to text-to-image diffusion models. In *IEEE International Conference on Computer Vision*. 3836–3847.
- [42] Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, and Jun-Yan Zhu. 2022. SDEdit: Guided Image Synthesis and Editing with Stochastic Differential Equations. In *International Conference on Learning Representations*.
- [43] Omri Avrahami, Dani Lischinski, and Ohad Fried. 2022. Blended diffusion for text-driven editing of natural images. In *IEEE Conference on Computer Vision and Pattern Recognition*. 18208–18218.
- [44] Zhenhua Yang, Dezhi Peng, Yuxin Kong, Yuyi Zhang, Cong Yao, and Lianwen Jin. 2024. FontDiffuser: One-shot font generation via denoising diffusion with multi-scale content aggregation and style contrastive learning. In *AAAI Conference on Artificial Intelligence*, Vol. 38. 6603–6611.
- [45] Rodolfo P dos Santos, Gabriela S Clemente, Tsang Ing Ren, and George DC Cavalcanti. 2009. Text line segmentation based on morphology and histogram projection. In *International Conference on Document Analysis and Recognition*. IEEE, 651–655.
- [46] Yipeng Sun, Jiaming Liu, Wei Liu, Junyu Han, Errui Ding, and Jingtuo Liu. 2019. Chinese street view text: Large-scale Chinese text reading with partially supervised learning. In *IEEE International Conference on Computer Vision*. 9086–9095.
- [47] Chenxia Li, Weiwei Liu, Ruoyu Guo, Xiaoting Yin, Kaitao Jiang, et al. 2022. PP-OCRv3: More attempts for the improvement of ultra lightweight OCR system. *arXiv preprint arXiv:2206.03001* (2022).
- [48] Yanjun Ma, Dianhai Yu, Tian Wu, and Haifeng Wang. 2019. PaddlePaddle: An open-source deep learning platform from industrial practice. *Frontiers of Data and Computing* 1, 1 (2019), 105–115.
- [49] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).
- [50] KoichiYasuoka. 2021. roberta-classical-chinese-large-char. <https://huggingface.co/KoichiYasuoka/roberta-classical-chinese-large-char>
- [51] Ethan-yt. 2020. GuwenBERT: A Pre-trained Language Model For Classical Chinese (Literary Chinese). <https://github.com/ethan-yt/guwenbert>
- [52] Boris Epshtein, Eyal Ofek, and Yonatan Wexler. 2010. Detecting text in natural scenes with stroke width transform. In *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2963–2970.
- [53] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. High-resolution image synthesis with latent diffusion models. In *IEEE Conference on Computer Vision and Pattern Recognition*. 10684–10695.
- [54] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in Neural Information Processing Systems* 30 (2017).
- [55] Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014).
- [56] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, et al. 2020. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *International Journal of Computer Vision* 128, 7 (2020), 1956–1981.
- [57] Ilya Loshchilov and Frank Hutter. 2019. Decoupled Weight Decay Regularization. In *International Conference on Learning Representations*.
- [58] Jonathan Ho and Tim Salimans. 2021. Classifier-Free Diffusion Guidance. In *NeurIPS Workshop on Deep Generative Models and Downstream Applications*.
- [59] JaidevAI. 2020. EasyOCR. <https://github.com/JaidevAI/EasyOCR>
- [60] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. 2018. The unreasonable effectiveness of deep features as a perceptual metric. In *IEEE Conference on Computer Vision and Pattern Recognition*. 586–595.
- [61] Haoqiang Fan, Hao Su, and Leonidas J Guibas. 2017. A point set generation network for 3d object reconstruction from a single image. In *IEEE Conference on Computer Vision and Pattern Recognition*. 605–613.
- [62] Jiaming Song, Chenlin Meng, and Stefano Ermon. 2021. Denoising Diffusion Implicit Models. In *International Conference on Learning Representations*.

A Training Dataset and Training Details

A.1 Dataset and Training Details of Restoration Preprocessor

Various denoising methods have been proposed to address similar levels of minor deterioration, including the GAN-based calligraphic image denoising method CIDG [2], which slightly outperforms other methods for minor noise removal from rubbing images. However, the CIDG training data consists mainly of very small spots of visual degradation similar to blue noise, which differ significantly from the real scratches and blotches in the deteriorated rubbing images. Consequently, the performance of the model is generally unsatisfactory when applied to real-world rubbing images, where deterioration is more pronounced. To enhance the model’s capability and generalization ability for rubbing images, we propose preparing a rubbing denoising dataset obtained from various sources for training from scratch, including synthetic ones and non-synthetic ones labeled by humans. The synthetic part of our dataset includes 800 image pairs in CIDG [2] and also 400 image pairs synthesized using a pipeline similar to that of our recognition training dataset (Section 4.1.3), but with a higher composition resolution at 256×256 . For real-world data, we collected 650 pairs of noisy real-world character images and their manually cleaned versions from RCRN [3]. The examples of all three data sources are illustrated in Fig. 15. We standardized all data pairs to 256×256 pixels. For training, we set the batch size to 8, the learning rate to $1e^{-4}$, and train the model from scratch for 135 epochs.

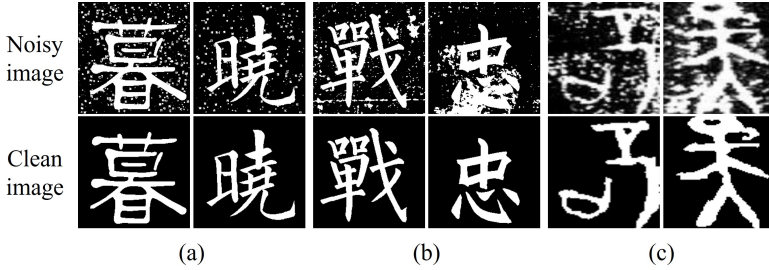


Fig. 15. Examples of training data used for training the restoration preprocessor. Samples are collected from (a) CIDG [2]; (b) data synthesis in the character recognition stage; (c) RCRN [3].

A.2 Details of Sampling Process in Rubbing Restoration Pipeline

During the sampling process, we apply the classifier-free guidance strategy to enhance the influence of priors $id(x)$ and x_f . For unconditional generation, we set the latent after encoding the vectors $id(x)$ and x_f to zero. Our sampling strategy can be formulated as:

$$\tilde{\epsilon}_{\theta}(z_t, t, C) = \epsilon_{\theta}(z_t, t, 0) + s(\epsilon_{\theta}(z_t, t, C) - \epsilon_{\theta}(z_t, t, 0)), \quad (5)$$

where s is the guidance scale and is set to 3, and $C = \text{concat}(E_{id}(id(x)), E_f(x_f))$. To speed up the sampling process, we use the DDIM sampler [62] with a sampling step of 25.

B Hyperparameters of Latent Diffusion Model

We provide a detailed explanation of the hyperparameters for our latent diffusion model in Table 4.

Table 4. Hyperparameters for the conditional latent diffusion model. Our model is trained on two NVIDIA RTX 3090 GPUs.

Downsample factor	8
z-shape	$16 \times 16 \times 4$
Diffusion steps	1000
Noise schedule	linear
Channels	128
Depth	3
Channel multiplier	1, 2, 3, 4
Number of heads	4
Batch size	64
Iterations	600,000
Learning rate	1e-4
Conditioning	Cross Attention (CA)
CA-resolutions	16, 8, 4
Embedding dimension	768
Transformer depth	1

Table 5. Recognition accuracy (%) with respect to maximum iteration counts for the recognition module.

		Minor Deterioration	Severe Deterioration	Overall
No MLM integration (OCR Only)		87.4	74.8	79.9
MLM-integrated with at most	1 iteration	93.2	82.2	86.7
	2 iterations	93.7	84.3	88.1
	3 iterations	93.8	84.3	88.2
	6 iterations	93.7	85.5	88.8

C Ablation Results

C.1 Necessity of Iterative Visual-context Integration

To investigate the importance of our visual-context integration approach, we conduct an ablation experiment comparing performance with and without integration of the masked language model. As shown in the first two rows of Table 5, adding the masked language model as guidance significantly improves recognition performance. The performance gain is more pronounced on severely deteriorated characters, demonstrating the usefulness of our context integration provided by the masked language model. However, in cases where OCR recognizes characters with consecutive errors, a single iteration of visual-context integration may sometimes fail to yield completely accurate results. In such situations, employing our iterative strategy, where the previous output is fed into the language model for the next round of visual-context integration, results in more accurate outcomes. As shown in Fig. 16, our iterative strategy gradually eliminates errors per character during the iterations, thanks to the contextual understanding of the masked language model. The statistics in Table 5 also support the necessity of our iterative approach.

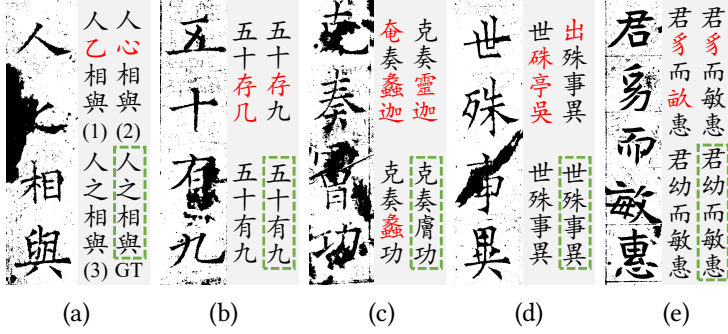


Fig. 16. Improvement of recognition accuracy with our proposed iterative strategy. Recognition results are placed on the right side of the rubbing input from (1) to (3): (1) raw OCR outputs, (2) first iteration outputs, (3) second iteration outputs. Incorrect predictions are highlighted in red while ground truths are wrapped by green dashed lines. Credits: (a, d) *Lanting ji*. Tang Dynasty, Chinese; (b, c) *Guo Yun mu zhi*. Tang Dynasty, Chinese; (e) *Kou Yinze mu zhi*. Northern Wei Dynasty, Chinese. Source: [Chinese Rubbings Collection - CURIOSity Digital Collections](#).

Table 6. Ablation study on the use of the font and identity priors. Bold for optimal values.

Methods	L1 ↓	RMSE ↓	LPIPS ↓	CD ↓	OCR% ↑
w/o identity prior	0.0720	0.2409	0.1648	2.89	31.07
w/o font prior	0.0541	0.2043	0.1211	2.12	91.72
Ours (w/o finetuning)	0.0538	0.2017	0.1185	2.10	91.76

C.2 Necessity of the Conditions in the Restoration Module

To understand the impact of different conditions on the performance of our restoration module, we conduct an ablation study. We first explore the importance of the font prior and identity prior. We remove the font prior by duplicating the 384-dimensional identity prior to form a 768-dimensional condition vector. The removal of the identity prior can be conducted in a similar manner. We then train the model with different prior configurations from scratch with the same number of iterations (without further finetuning) and compare the results using our restoration test dataset. Fig. 17 shows that without a reference image as the font prior, the model may render strokes with styles that are not consistent with the character’s designated font. While the diffusion model may use the undeteriorated parts as an additional font reference, its effectiveness depends on the complexity of the ground truth strokes in the deteriorated regions (the 1st and 3rd rows in the figure). The statistics also suggest a minor performance drop, as shown in Table 6. On the other hand, when the character identity prior is removed in the condition, the restoration process generates random, meaningless strokes (the third column in Fig. 17), leading to a severe decline in relevant evaluation metrics, especially OCR accuracy.

C.3 Necessity of the Deteriorated Image as the Input

We explore the importance of using deteriorated images as input. Notably, without using the deteriorated image as input, our method degenerates into character image generation, starting from Gaussian noise based solely on the character identity and the reference image. As shown in the red box in the second row of Fig. 18, omitting deteriorated character images makes it impossible



Fig. 17. Ablation study on the use of the font and identity priors.

to preserve the undeteriorated parts of the image. However, using the deteriorated image as input effectively resolves this issue.



Fig. 18. Ablation study on the use of the deteriorated image as input. The case without image input is the font generation task, which only uses text as input.

C.4 Necessity of Finetuning VAE Model

We highlight the importance of using character images to finetune the kl-f8 VAE model from [53]. We compare two restoration modules: one trained on the generic pre-trained VAE and the other on our finetuned VAE. We generate multiple character images with the same font and identity prior for both models, as illustrated in Fig. 19. Since the generic pre-trained VAE is trained on numerous natural photographs, which differ significantly from our character data with sharper



Fig. 19. Ablation study on the effect of finetuning on the original VAE model.

boundaries, using the VAE without finetuning leads to noticeable artifacts (see the first row of Fig. 19), significantly affecting the visual quality.