Chinese Character Inpainting with Contextual Semantic Constraints

Jiahao Wang
wjh849583369@tju.edu.cn
College of Intelligence and Computing
Tianjin University

Gang Pan*
pangang@tju.edu.cn
College of Intelligence and Computing &
School of New Media and Communication
Tianjin University

Di Sun*
dsun@tust.edu.cn
Tianjin University of Science and Technology
Tianjin University

Jiawan Zhang
jwzhang@tju.edu.cn
College of Intelligence and Computing
Tianjin University

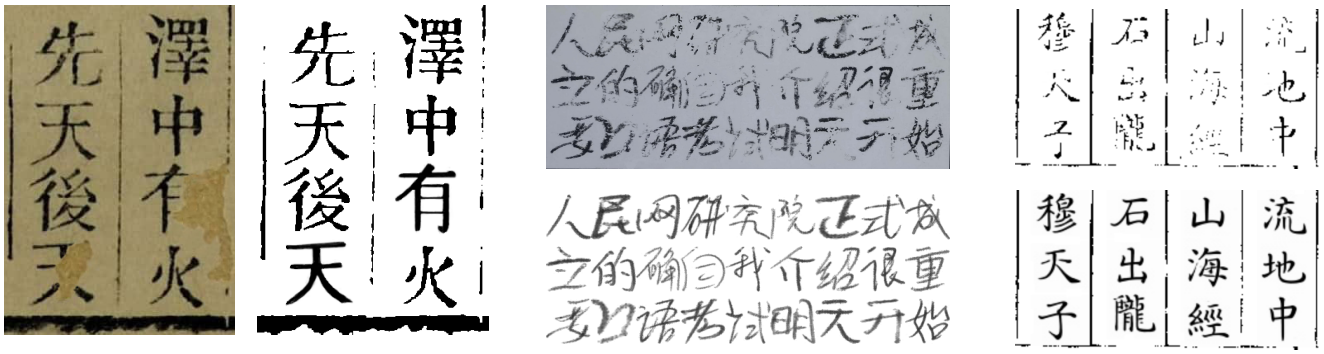

Figure 1: Chinese character inpainting results by the proposed method. These examples are taken from different scene texts. Although some characters are almost unrecognizable, our method can still infer and restore reasonable characters based on contextual semantics.

ABSTRACT

Chinese character inpainting is a challenging task where large missing regions have to be filled with both visually and semantic realistic contents. Existing methods generally produce pseudo or ambiguous characters due to lack of semantic information. Given the key observation that Chinese characters contain visually glyph representation and intrinsic contextual semantics, we tackle the challenge of similar Chinese characters by modeling the underlying regularities among glyph and semantic information. We propose a semantics enhanced generative framework for Chinese character inpainting, where a global semantic supervising module (GSSM) is introduced to constrain contextual semantics. In particular, sentence embedding is used to guide the encoding of continuous contextual characters. The method can not only generate realistic Chinese character, but also explicitly utilize context as reference during network training to eliminate ambiguity. The proposed method is evaluated on both handwritten and printed Chinese characters with various masks. The experiments show that the method successfully predicts missing character information without any mask input, and achieves significant sentence-level results benefiting from global semantic supervising in a wide variety of scenes.

CCS CONCEPTS

• **Computing methodologies** → **Adversarial learning**; **Image processing**; **Natural language processing**.

KEYWORDS

character inpainting; contextual semantics; global semantic supervising; sentence embedding

ACM Reference Format:
Jiahao Wang, Gang Pan, Di Sun, and Jiawan Zhang. 2021. Chinese Character Inpainting with Contextual Semantic Constraints. In *Proceedings of the 29th ACM International Conference on Multimedia (MM '21), October 20–24, 2021, Virtual Event, China.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3474085.3475333

*Corresponding authors.

MM '21, October 20–24, 2021, Virtual Event, China

ACM ISBN 978-1-4503-8651-7/21/10…$15.00
https://doi.org/10.1145/3474085.3475333

1 INTRODUCTION

Chinese character, either handwritten or printed, is the basic unit of representing text information for communication and collaboration. The original carriers of Chinese characters are ancient documents or stone inscriptions. However, some parts of the Chinese characters may be damaged due to improper storage or aging factors. We aim to fill the missing regions in Chinese character images with accurate synthesized contents, and call this process as Chinese character inpainting (see Figure 1). Restoring these characters contributes in enhancing text documents visually and making them easier to be recognized. Thus, it is potential to support a variety of text-related tasks, such as text recognition and text classification. Meantime, it is of great historical significance and cultural value.

Despite a lot of researches on natural image synthesis, there are few studies on character inpainting, especially for ideograms like Chinese characters. Unlike English characters, which consist of 26 alphabets, the number of Chinese characters is quite large. Moreover, each Chinese character is a kind of pictographic symbol with a complex and unique structure, which makes the restoration of Chinese character images more difficult. Existing image inpainting techniques [7, 20, 22, 33] synthesize an image by filling plausible contents in the missing regions. With the development of deep learning, recent CNN-based works on image inpainting have shown promising results, most of which follow the encoder-decoder structure. These methods may obtain visually plausible character structures, but often generate blurry non-existence Chinese characters (Figure 2(a)). In addition, masks indicating the missing regions are usually required as input, which may ultimately limit the utility of these methods in character inpainting.

Different from natural images, the basic unit of Chinese characters is the stroke, which is composed of various exact lines that has a specific shape called a glyph. As a result, some approaches regard the character inpainting task as a deteriorated line drawing restoration problem [25, 26]. These approaches attempt to complete discontinuous stroke after automatic gap detection from a character. Due to the diversity and complexity of Chinese characters, these approaches are effective when the stroke structure is clear and unambiguous, but may fail to deal with complex structure with multiple lines and intersections, resulting in pseudo characters(Figure 2(b)). To synthesize more realistic results, some researches [4, 15] employ generative adversarial networks (GAN) to restore certain Chinese character. These methods do not take into account the contextual semantics(Figure 2(c)), although some certain legal characters can be generated, they do not conform to the actual context. In addition, large missing regions are also challenging for such approaches because it is difficult to identify stroke structure without the constraint of character glyph.

The glyph representation of Chinese characters encode rich information of their meanings. It is intuitive that the glyph representation and semantic information of a character should benefit from each other in character-related computer vision or natural language processing tasks. More recently, there have been some efforts applying visual glyph feature to language understanding tasks, including text classification [13, 35], word analogy and word similarity [29]. These works verify the correlation between glyph information and semantic information of character images. Moreover, some works

Figure 2: Illustration of Chinese character inpainting cases with (a) blurry characters, (b) pseudo characters, (c) ambiguous characters, (d) correct characters.

attempt to introduce text features into some computer vision tasks from the perspective of embedding [14, 21, 23, 24]. These methods prove that embedded semantic information is helpful to the text related computer vision tasks, but they deal with English characters and can only restore a single word.

In this paper, we propose a contextual semantics enhanced generative framework for Chinese character inpainting. Our method can automatically recognize the missing regions in a character or short text image, and adaptively restore the missing stroke of each character. First, we design an inpainting model that consists of an encoding-decoding generator and a discriminator to synthesize the missing contents. Second, we introduce a sub-network structure named global semantic supervising module (GSSM) to restrict glyph representation. The key idea is to jointly conduct character visual feature and contextual semantic information in the inpainting process. Sentence embedding from NLP is regarded as a contextual reference to guide the encoding features. Specifically, we obtain the sentence embedding from a pre-trained language model and the semantic information from GSSM respectively, then compute a loss between them during training. More concretely, all characters in a sentence are intercepted to form a batch of images as input. These character images are considered as a whole in the processing of GSSM, but they do not affect each other in the generator and discriminator. The semantic module can predict the semantic information of a set of character images that represent a sentence(Figure 2(d)).

In summary, the main contributions are as follows:

- We offer a new perspective into Chinese character inpainting with contextual semantics. The combination of image and text provides richer semantics, allowing the model to infer accurate Chinese characters.
- We propose a network with global semantic supervising module (GSSM), which takes sentence embedding from NLP as contextual constraints to optimize the glyph features of each Chinese character.

- We contribute a new handwritten and printed image dataset with semantic labels for Chinese character image inpainting, in the hope that it will kick-start research effort on solving this challenging problem.

2 RELATED WORK

Image inpainting approaches can be roughly divided into two categories: traditional methods based on diffusion or patch matching, and recent methods based on deep learning. Diffusion-based methods [1, 3], which are restricted to locally available information, propagate neighboring information into the missing regions. PatchMatch [2] is a typical patch based method which finds similar blocks on the original image and fill them in the inpainting position. These methods are feasible for repairing local distortions such as texture damage, but they are insufficient to deal with some global information distortions. Recently, many image inpainting tasks based on the deep learning apply the visual and semantic information from the undamaged regions of the original image to infer the filling content, where the contextual features [20], content features [33] and GAN features [22] are formulated to optimize the results.

Character inpainting is more inclined to restore local lines from the perspective of visual features. The methods in [25] and [18] work well for phonogram but are not suitable for ideogram with with complex structure such as Chinese characters. Subsequently, some approaches attempt to introduce the glyph features. Chang et al. [4] propose a hierarchical learning network named HAN to extract hierarchical features to reconstruct the damaged strokes of printed Chinese character. Li et al. [15] pose a generative adversarial network combining with contextual loss to improve the inpainting effect of handwritten Chinese character. However, the above methods often generate some pseudo-Chinese characters composed of strokes that seem reasonable but do not actually exist. Due to the lack of interaction between adjacent characters in the text image, these methods are likely to produce ambiguous characters.

It is obvious that only relying on scattered glyph information cannot guarantee to infer the correct character. Intuitively, the glyph representation and semantic information of a character should complement each other. In recent years, the interaction of visual and semantic information has shown great progress in multi-modal tasks such as machine translation, image annotation, and cross-modal retrieval. On the one hand, existing research has introduced glyph information into text classification, and has achieved good results by converting high-dimensional character glyph information into low-dimensional vectors as glyph embedding.

Among these, Wilkinson et al. [32] try to embed image features into a word embedding space for text spotting. Zhang et al. [35] utilize Chinese character representations for text classification. Su et al. [29] find that the glyph embedding improves both word analogy and word similarity results. The CE-CLCNN [13] applies damaged character image to classify text. The Glyce method [17] combines glyph embeddings with BERT embeddings [9] for a variety of Chinese NLP tasks. As mentioned before, the state-of-the-art methods using glyph embedding emphasize the glyph structure to enhance semantic information. However, these related studies do not show the effect of semantic information on the improvement of glyph structure.

On the other hand, text features also play an important role in computer vision tasks from the perspective of embedding. STEFANN [24] is designed to modify the text content in an image at character-level, which uses label embedding to represent and edit each character. Some cross-modality tasks [14, 23] try to learn feature embedding from word images and text labels for word spotting and recognition tasks. Recently, the idea of SEED [21] is to use word embedding from a pre-trained language model to make up for the lack of contextual semantic information. In SNR [34], global semantic reasoning module is introduced to capture global semantic context for accurate scene text recognition. These studies show that semantic modules that process contextual semantic information can be applied in image processing tasks.

3 APPROACH

In this section, we describe the proposed framework for Chinese character inpainting. Our goal is to synthesize the missing contents that are both visually realistic and semantically consistent with the whole text. Figure 3 shows the proposed framework that consists of a generator, a discriminator, GSSM and a pre-trained language module. The generator is used to extract visual features and synthesize the missing contents. The discriminator can improve the quality of synthesized results, especially for the stroke details. The GSSM can capture sentence semantic features, which guides the generator to synthesize contextually coherent characters. The pre-trained language module serves as a constraint on the information predicted by the semantic module.

3.1 Generator

The generator G follows the architecture in HAN network [4], which is designed as an encoder(G_{en})-decoder(G_{de}) structure. Skip connections between mirrored layers in the encoder and decoder stacks have been added to the generator. It is worth mentioning that the skip connection can restore more details. In order to retain more spatial support to generate the masked regions, we use convolution with 2×2 pixel strides and convolutional kernel for down-sampling rather than pooling operation. Additionally, the framework with more convolutional layers has better performance to restore line drawings so that more uniform-sized convolutional layers are added to extract enough local features. Instead of a relu activation used in the output layer in HAN, we use a sigmoid activation to map it to (0,1) in the output.

Let $\tilde{x}$, x, n be input images, ground truth images, and batch size, respectively. The input of the encoder is represented as $\{\tilde{x}_1, \tilde{x}_2, ..., \tilde{x}_n\}$ and the output of the encoder is a batch of feature sequences f:

$$f = G_{en}(\tilde{x}) \tag{1}$$

corresponding to $\{f_1, f_2, ..., f_n\}$. Feature sequences f have two functions, one is to predict the semantic information by GSSM and the other is as the input of the decoder to predict the character image x_{pred} for the masked region

$$x_{pred} = G_{de}(f). \tag{2}$$

The generator aims to synthesize characters similar to the specified ground-truth ones. L1- or L2-norm are often used to measure the pixel distance between paired images. Considering the impact

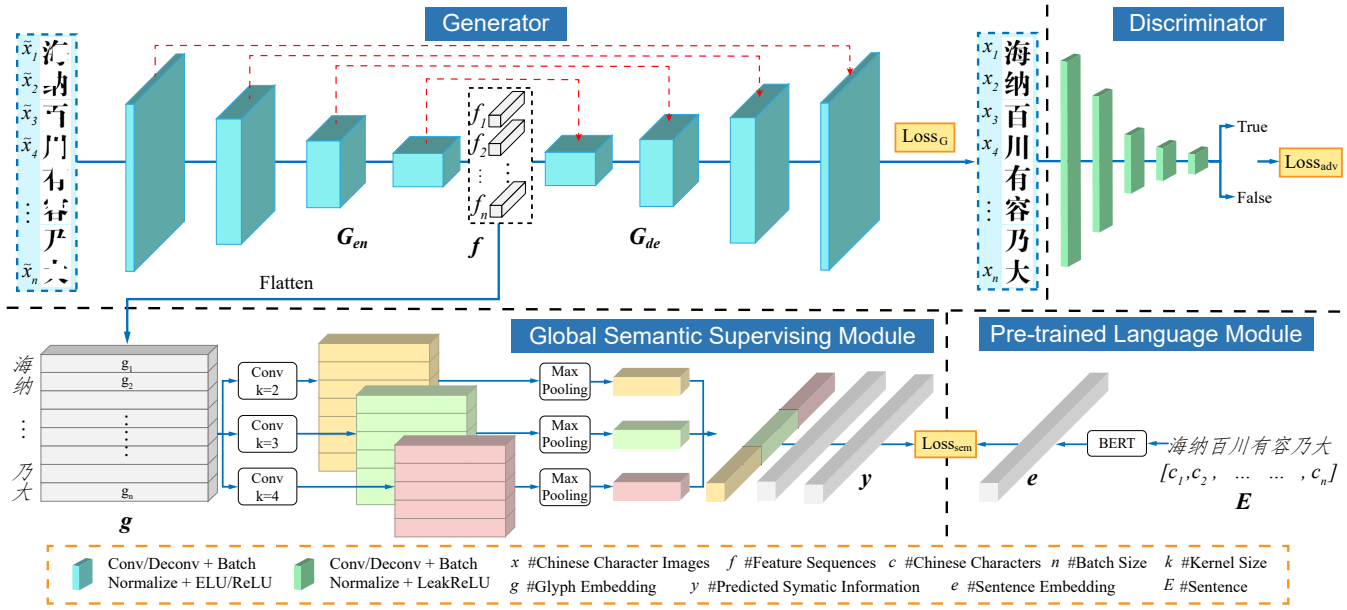

Figure 3: Framework Overview. It consists of a generator, a discriminator, a global semantic supervising module (GSSM) and a pre-trained language module. The generator is based on encoder-decoder. The encoder follows the Conv-BatchNorm [12] - ELU [6] architecture. The decoder follows the Conv-BatchNorm-ReLU except for the last layer, which uses a sigmoid activation function to keep the output in [0,1] range. A batch of incomplete images are fed to the encoder to generate feature maps *f*, which are inputs to the decoder and GSSM respectively. Predicted images from the decoder are fed to the discriminator following Conv-SpectralNorm [19]-LeakyReLU architecture.

of multiple typefaces, we further introduce a per-pixel reconstruction loss $\mathcal{L}_{rec}$ to the generator, which is the L1 distance between the network output and the ground-truth image. The $\mathcal{L}_{rec}$ loss is necessary to evaluate details and structure differences

$$\mathcal{L}_{rec}(G) = \|\phi(x) - \phi(x_{pred})\|_1, \tag{3}$$

where ϕ is the pre-trained 16-layer VGG network [28]. Furthermore, a content loss $\mathcal{L}_{con}$ is used to represent the difference of the predicted regions with respect to the ground-truth. It is defined by

$$\mathcal{L}_{con}(G) = \|x \odot x_{pred}\|_2^2, \tag{4}$$

here $\odot$ is the pixel-wise multiplication and $\|\cdot\|_2$ is the Euclidean norm. Therefore, the generative loss can be defined as

$$\mathcal{L}_G = \mathcal{L}_{con} + \lambda_1 \mathcal{L}_{rec}. \tag{5}$$

3.2 Discriminator

Although the generator network can capture missing information, it does not ensure that the filled region is visually realistic and coherent. To encourage more realistic results, we adopt a discriminator D that serves as a binary classifier to distinguish between real and fake images. The goal of this discriminator is to help improve the quality of synthesized results such that the trained discriminator is fooled by unrealistic images. Our discriminator structure is similar to DCGAN [22], using Leakly ReLU activation function with slope 0.2 after each convolution layer. Spectral normalization [19] is applied to stabilize training. We use a sigmoid activation to map the result to [0,1] in the output, in which 0 means fake and 1 means real.

We use x_{pred} and x as input that predicts whether or not a character image is real. The objective of a GAN can be formulated as

$$\mathcal{L}_{adv}(G, D) = \mathbb{E}_{(x)} \log(D(x)) + \mathbb{E}_{(x_{pred})} \log(1 - D(x_{pred})), \tag{6}$$

where the generator G is trained to minimize this objective against the adversarial D that tries to maximize it.

3.3 Global Semantic Supervising Module

In this subsection, we first address how we integrate global semantic supervising module (GSSM) into our unified inpainting network, and then discuss details of the contextual module.

Each of feature sequences $\{f_1, f_2, ..., f_n\}$ is flattened into h dimensional glyph vectors $\{g_1, g_2, ..., g_n\}$ that are similar to pre-trained word vectors as embedding layers. These visual-to-semantic information contains the visual features of continuous n characters with semantic coherence. Let g be all glyph vectors applying concatenation operator $\oplus$

$$g = g_1 \oplus g_2 \oplus ... \oplus g_n \tag{7}$$

with the shape of $n \times h$, which are input to GSSM to predict a feature vector. We uses filter w_k with the kernel size of k and 256 channels to obtain multiple features by one-dimensional convolution. It's represented as

$$v(k) = p_{n-k+1}(\sigma(w_k g + b)). \tag{8}$$

Figure 4: Comparison with the state-of-the-art. A, B, C, D are sentences consisting of eight characters, respectively. These images cover handwritten and printed characters with regular or irregular masks. In addition to the correctness of strokes or radicals, our method avoids generating ambiguous characters with high visual credibility. Even if the characters are almost completely damaged, our network can still infer the correct characters based on the contextual semantics.

Here σ is a ReLU activation function, b is a bias term and p is max-pooling operation with the kernel size of $n-k+1$. Each of feature vectors $v(k)$ is concatenated to form a new feature vector v

$$v = v(2) \oplus v(3) \oplus v(4), \tag{9}$$

where $\oplus$ denotes the concatenation operator. We then apply two linear functions that randomly dropouts [10] arbitrary elements to prevent overfitting, and the semantic information y is predicted as following:

$$y = w_2\sigma(w_1 v + b_1) + b_2. \tag{10}$$

Here w_1, w_2, b_1, b_2 are weights. ReLU activation function and Batch Normalization are used for convolutional layers. Dropout rate of linear functions is set to 0.5.

The GSSM is also supervised by a kind of loss, called the semantic loss. The semantic loss is used to optimize the completion of contextual components. It mainly considers how to use appropriate text representation methods with semantics to compute semantic similarity. It is defined as follows:

$$\mathcal{L}_{sem} = 1 - \cos(y, e) \tag{11}$$

where $\mathcal{L}_{sem}$ is the cosine embedding loss between the predicted semantic information y and the sentence embedding e generated by our pre-trained language module.

3.4 Pre-trained Language Module

To supervise the predicted semantic information, we choose BERT model as our pre-trained language module, which is based on 12 transformer layer structure [31]. The embedding vector is then produced by averaging from the hidden layers of each token from the second layer to the last layer. The module is designed to pre-train deep bidirectional representations from unlabeled text by jointly conditioning on both left and right context in all layers. As a result, the pre-trained BERT model can be finetuned with just one additional output layer to create state-of-the-art models for a wide range of tasks. In our approach, a sentence $E = \{c_1, c_2, ..., c_n\}$ is encoded by the Chinese pre-trained BERT to get sentence embedding e, where each character c_i corresponds to the content of each character image x_i.

We train the framework with a special designed hybrid loss $\mathcal{L}$, which is the combination of generative loss $\mathcal{L}_G$, adversarial loss $\mathcal{L}_{adv}$ and semantic loss $\mathcal{L}_{sem}$. As shown in Figure 3, they correspond to the generator, discriminator and GSSM respectively. The overall loss function is defined as follow:

$$\min_G \max_D \mathcal{L} = \min_G \left(\mathcal{L}_{con} + \lambda_1 \mathcal{L}_{rec} + \lambda_2 \max_D(\mathcal{L}_{adv}) + \lambda_3 \mathcal{L}_{sem} \right) \tag{12}$$

where λ_1, λ_2 and λ_3 are hyperparameters that balance the contribution of different losses.

Table 1: Comparison of different methods in terms of PSNR, SSIM, OCR confidence and accuracy. The values in bold indicate the best performance.

Dataset	Methods	PSNR	SSIM	CRNN-Confidence	CRNN-OCR	DenseNet-Confidence	DenseNet-OCR
Printed	EBII	13.193752	0.715668	0.221428	0.105678	0.285756	0.146688
	HCCGAN	20.218434	0.863805	0.679080	0.717391	0.748061	0.836232
	LDFCN	24.448020	0.893142	0.715833	0.779710	0.809689	0.871159
	HAN	24.078704	**0.902420**	0.713677	0.784058	0.821934	0.892754
	Ours	**27.188082**	0.898676	**0.725509**	**0.800000**	**0.824841**	**0.901449**
	Ground Truth	-	-	0.728168	0.804348	0.827814	0.905797
Handwritten	EBII	5.348093	0.077376	0.178904	0.173594	0.168758	0.136952
	HCCGAN	18.816881	0.818332	0.513416	0.630284	0.568365	0.621988
	LDFCN	25.847255	0.931537	0.663934	0.791689	0.709689	0.787474
	HAN	**26.777639**	**0.943565**	0.766867	0.842694	0.781934	0.837791
	Ours	26.698042	0.942825	**0.797347**	**0.881394**	**0.804841**	**0.850243**
	Ground Truth	-	-	0.817668	0.901798	0.835935	0.884479

4 EXPERIMENTAL RESULTS

4.1 Datasets

To train the whole network, a dataset that contains Chinese character images with text label is required, but this kind of dataset is not common. There is no even public data set available for Chinese characters inpainting. To this end, we enrich **CASIA-HWDB1.1**[1], a widely used handwritten Chinese character dataset, with various masks. We also create a new printed dataset, based on synthetically generated printed Chinese character images. Both are associated with the text label. The datasets can be found in the page[2].

Label for Chinese Character Images (LCCI): As mentioned earlier, Chinese character inpainting should be targeted at a meaningful text rather than a single Chinese character. And the proposed method is an end-to-end learning model, that is, the input of the generator are a group of continuous damaged character images from a sentence, and the output are a set of completed character images. So the key problem is how to find the corresponding label for each Chinese character in a text image. **THUCTC** dataset [30] is widely used for the NLP tasks such as text classification. We take about 50,000 text data from **THUCTC**. After pre-processing, a text dataset **LCCI** consisting of 100,000 sentences with 634 Chinese characters is obtained. The number of characters in each sentence is kept in the range (7, 32). Inspired by word embedding, we use a pre-trained language model named Chinese BERT-WWM [8] to perform a "sentence embedding" process to generate some sentence vectors. The size of each vector is set to 768 by default. These selected labels are used to associate images in reverse.

Handwritten Chinese Character Data (HCCD): **CASIA-HWDB1.1** contains 1,172,907 handwritten Chinese character images and each category is written by 300 people. For dataset **HCCD**, 190,220 images are picked from **CASIA-HWDB1.1** corresponding to 634 Chinese characters in dataset **LCCI**, each of which contain 300 pairs of grayscale images. These images are normalized to 64×64 size. For each pair, one with the mask is the input image and the other is the target image. We use two types of image masks: regular and irregular. Regular masks that are rectangular account for 25% of total image pixels. Irregular masks [16] cover the central regions of characters. We then split the dataset **HCCD** into the training set, validation set and testing set with 6:2:2 ratio.

Printed Chinese Character Data (PCCD). Motivated by the CE-CLCNN method [13], we similarly design a printed Chinese character image dataset **PCCD**. Each of 634 Chinese characters from dataset **LCCI** is associated with 1000 pairs of grayscale images, and we use the same method as the handwritten dataset **HCCD** to perform data pre-processing. Moreover, 36 types of printed fonts are used to increase the generalization ability of the network.

Examples of different masks are shown in Figure 5. Compared with the common image inpainting method, our method does not need to take masks indicating the missing regions as the input.

Figure 5: Examples of random masks

4.2 Implementation Details

To alleviate the long-tailed data distribution problem, we apply two strategies consisting of content training and semantic context training. For content training, we only train the generator and discriminator, and randomly select image pairs from the dataset **HCCD/PCCD** as training data, where batch size n, λ_1, λ_2 and λ_3 are set to 32, 0.5, 0.001, 0 respectively. The role of content training is to ensure that all kinds of characters in dataset are fully trained.

For semantic context training, sentence labels in **LCCI** are used as indexes to find the corresponding character images in **HCCD/PCCD**. These images are used as training batch, to ensure the

[1] http://www.nlpr.ia.ac.cn/databases/handwriting/Home.html
[2] https://github.com/WANGJH9953/CCID2021

context coherence of multi-word images. The corresponding parameters λ_1, λ_2 and λ_3 for the inpainting network with semantic supervising module are set to 0.5, 0.001, 1, respectively. We set up the semantic context training for 10 epochs after executing the content training for 30 epochs. Furthermore, handwritten dataset and printed dataset are trained independently.

For other real data, we perform some preprocessing operations to convert it into inputable characters. We first use EAST [36] to mark the text regions and then apply the MSER [5] algorithm to detect the binary masks of individual characters. In fact, the main difference for a Chinese character (after propocessing) is the font style. Our model can automatically match the same or similar font style and use it as a basis for predicting and filling in missing strokes.

The proposed network is implemented in PyTorch, which adopts Adam as optimizer. The learning rate is set at 0.0001 for both generator and GSSM, and 0.00001 for discriminator until losses level off. The operating system is Ubuntu18.04 running on Intel Core i7-10700K CPU. The network training process is performed on RTX 2080Ti GPU with 11GB GPU memory and 16GB RAM. The training time varies from 27 hours to 35 hours on different datasets.

4.3 Qualitative Results

To validate the proposed method, we compare the inpainting performance with the previous state-of-the-art methods: the Exemplar-Based Image Inpainting (EBII) [7], the Fully Convolutional Network for Line Drawings (LDFCN) [25], the Generative Adversarial Network for Handwritten Chinese Character (HCCGAN) [15] and HAN [4]. Four sets of semantically coherent images with regular or irregular masks are compared by four methods.

It is worth noting that the incomplete regions are highlighted in grey in the paper. Actually, we don't need to mark the incomplete regions in the training and testing process. As illustrated in Figure 4, the traditional method EBII synthesizes unrecognizable characters due to the large mask. Other deep learning-based methods, such as HCCGAN and LDFCN, significantly improve the generative results, but still produce blurry or even disordered strokes. HAN uses hierarchical discriminator to make the characters clear but it is easy to generate pseudo or ambiguous characters. Our method can accurately capture visual information, so as to avoid the phenomenon of gaps or local blurring for a single Chinese character as much as possible. Especially for printed Chinese characters, even if the fonts are different, our method can still maintain the glyph structure.

When ignoring the global semantics, characters with similar glyph structures are easily confused during the inpainting process. Relying only on the information of a single image is likely to output similar ambiguous characters. In our work, the semantic module utilizes pre-trained sentence embeddings as constraints to supervise the content of all character images. It is worth noting that even if a single character is almost completely occluded, our method can still learn the semantic information to output the correct character.

4.4 Quantitative Evaluation

Like other image synthesis tasks, image inpainting lacks good quantitative evaluation metrics. Nevertheless we follow the previous image inpainting works by reporting Peak Signal to Noise Ratio (PSNR) and Structural Similarity (SSIM) for visual evaluation criterion, as shown in Table 1. Compared with the results of PSNR + SSIM, our final results has comparable performance, because PSNR and SSIM may struggle to distinguish high similarity images such as pseudo or ambiguous characters. Therefore only relying on these two evaluation metrics may not be an effective metric. In order to evaluate our method more effectively, we further adopt Optical Character Recognition (OCR) confidence and OCR accuracy for semantics assessment. DenseNet [11] and CRNN [27] are selected for OCR quantitative indicators. The final results after 40 epochs are shown in Table 1. The proposed method achieves better, according to OCR accuracy and confidence.

Table 2: Results in terms of PSNR, SSIM, OCR confidence and accuracy for ablation study.

Methods	PSNR	SSIM	Confidence	OCR
content	25.3462	0.8937	0.7984	0.8726
content + rec	26.6089	0.9008	0.8122	0.8803
content + rec + adv	27.1881	0.9024	0.8248	0.8813
content + rec + adv + sem	27.7124	0.9043	0.8272	0.9014
ground truth	-	-	0.8278	0.9057

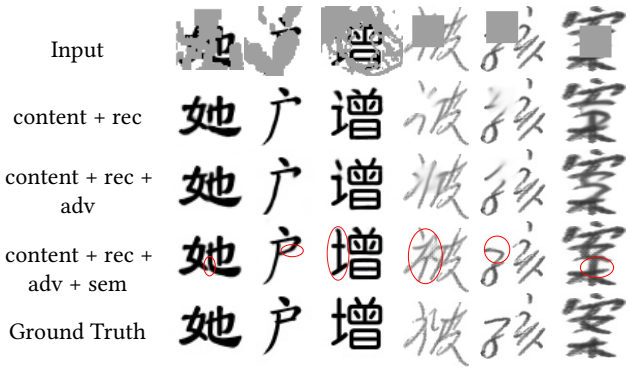

Figure 6: Ablation study. Some generated handwritten or printed characters are ambiguous although they have high visual credibility. The method with semantic module can correctly recognize the glyph structure so that the ambiguity can be reduced.

4.5 Ablation study

We now turn our attention to the key assumption of this task: contextual semantic information helps Chinese character inpainting. We analyze how GSSM contributes to the final performance of image inpainting. We also show the influence of different factors of the proposed network by adding/removing corresponding loss functions. Figure 6 illustrates the results with different fonts and masks. Content loss and reconstruction loss in the second row enable to maintain the local continuation of thickness and curvature, which can correctly restore the structure of strokes for line ends. Adversarial loss in the third row improves the relative position of strokes, which makes full use of glyph visual information to

reduce the appearance of pseudo characters. Semantic loss at the bottom row improves the correctness of neighbor characters from global context semantic constraints. The quantitative results are illustrated in Table 2. The comparison indicates that contextual semantic guidance is a crucial part to the success of our model.

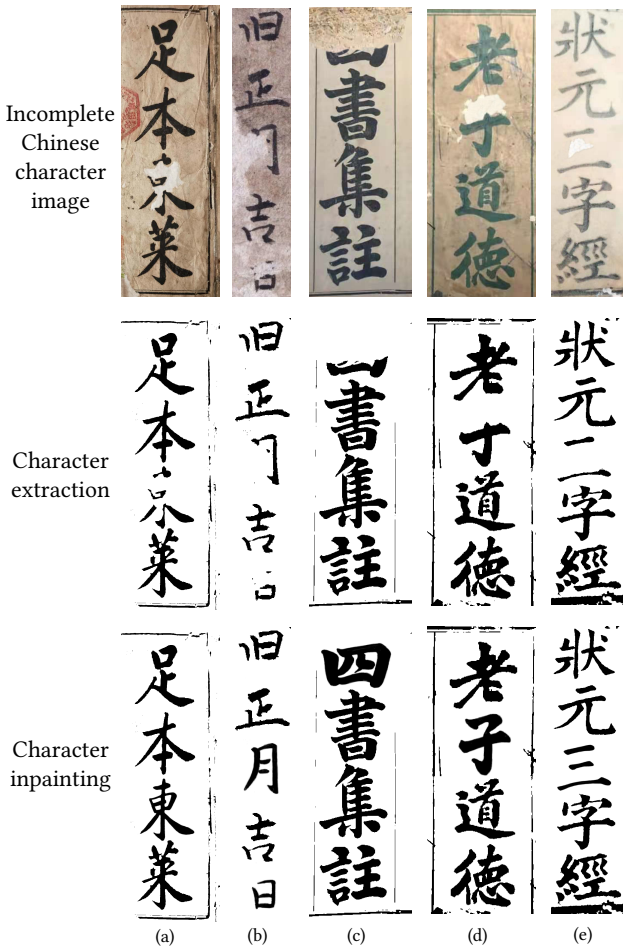

Figure 7: Inpainting results on incomplete characters from ancient books. Note that the bottom row(e) indicates the result "Three-Character Classic", rather than "Two-Character Classic".

4.6 Application

We apply our model to ancient books reparation. As illustrated in Figure 7, our model is able to correctly reconstruct the missing part of one character. Especially in Figure 7(e), each extracted character is individually legal, but it is unreasonable to connect them together to form the "Two Character Classics". While our full model can reconstruct the correct content, that is, "Three Character Classics".

Our algorithm can be also employed to recover various types of incomplete texts, such as those caused by light reflection or paper damage. Under semantic supervision mode, our model learns to correctly synthesize the original characters according to the contextual semantics of the characters. Some examples are shown in Figure 8.

5 CONCLUSION

This paper demonstrates the effectiveness of introducing contextual semantic guidance into Chinese character image inpainting. We propose a contextual semantics enhanced inpainting scheme that combines the glyph features of the image and semantic information as the generation condition. The proposed method can restore incomplete Chinese character images while maintaining content information and style consistency, even if some characters are almost unrecognizable. Meanwhile, the contextual semantics module can largely avoid the generation of ambiguous characters with similar glyph representations. We also built a dedicated Chinese character image dataset including handwritten and printed for performance evaluation. The experimental results demonstrate that the proposed method outperformed compared models in subjective and objective comparisons, and the model outputs are contextual coherent. It should be pointed out that the restoration effect of Chinese characters is greatly affected by its semantics from pre-trained language module. For ancient literature, there is little semantic reference information available. Moreover, our method may fail if Chinese character images are badly damaged. Future work may consider inpainting task of other ideographic and phonetic scripts such as Japanese and English.

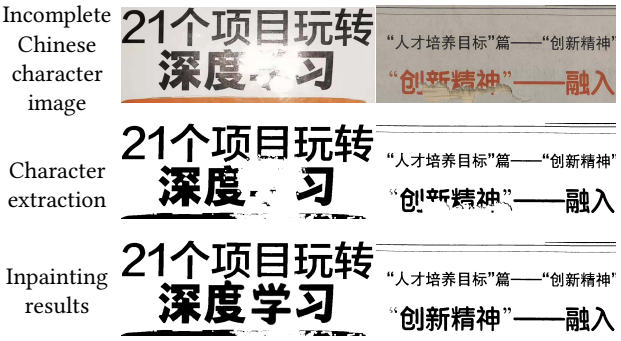

Figure 8: Inpainting results on incomplete characters caused by light reflectance or paper damage.

ACKNOWLEDGEMENTS

This work was supported by National Key Research and Development Program of China under Grant(No.2019YFC1521200). And we also thank for the support of NVIDIA Corporation with the donation of the GPU used for this research.

REFERENCES

[1] Coloma Ballester, Marcelo Bertalmio, Vicent Caselles, Guillermo Sapiro, and Joan Verdera. 2001. Filling-in by joint interpolation of vector fields and gray levels. *TIP* 10, 8 (2001), 1200–1211.
[2] Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B Goldman. 2009. PatchMatch: A randomized correspondence algorithm for structural image editing. *ToG* 28, 3 (2009), 24.
[3] Marcelo Bertalmio, Guillermo Sapiro, Vincent Caselles, and Coloma Ballester. 2000. Image inpainting. In *Siggraph*. ACM, USA, 417–424.

[4] Jie Chang, Yujun Gu, Ya Zhang, and Yan-Feng Wang. 2018. Chinese Handwriting Imitation with Hierarchical Generative Adversarial Network. In *BMVC*. BMVA Press, UK, 290.

[5] Huizhong Chen, Sam S Tsai, Georg Schroth, David M Chen, Radek Grzeszczuk, and Bernd Girod. 2011. Robust text detection in natural images with edge-enhanced maximally stable extremal regions. In *ICIP*. IEEE Computer Society, USA, 2609–2612.

[6] Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. 2016. Fast and accurate deep network learning by exponential linear units (ELUs). In *ICLR*. JMLR.org, USA, 1–14.

[7] Antonio Criminisi, Patrick Pérez, and Kentaro Toyama. 2004. Region filling and object removal by exemplar-based image inpainting. *TIP* 13, 9 (2004), 1200–1212.

[8] Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, Ziqing Yang, Shijin Wang, and Guoping Hu. 2019. Pre-training with whole word masking for chinese bert. *arXiv preprint arXiv:1906.08101* (2019).

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*. Association for Computational Linguistics, USA, 4171–4186.

[10] Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. 2012. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580* (2012).

[11] Gao Huang, Zhuang Liu, and Laurens Van Der Maaten. 2017. Densely connected convolutional networks. In *CVPR*. IEEE Computer Society, USA, 4700–4708.

[12] Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*. JMLR.org, USA, 448–456.

[13] Shunsuke Kitada, Ryunosuke Kotani, and Hitoshi Iyatomi. 2018. End-to-End Text Classification via Image-based Embedding using Character-level Networks. In *AIPR*. IEEE Computer Society, USA, 1–4.

[14] Praveen Krishnan, Kartik Dutta, and C. V. Jawahar. 2018. Word Spotting and Recognition Using Deep Embedding. In *IAPRW*. IEEE, USA, 1–6,.

[15] Jianwu Li, Ge Song, and Minhua Zhang. 2020. Occluded offline handwritten Chinese character recognition using deep convolutional generative adversarial network and improved GoogLeNet. *Neural Computing and Applications* 32, 9 (2020), 4805–4819.

[16] Guilin Liu, Fitsum A Reda, Kevin J Shih, Ting-Chun Wang, Andrew Tao, and Bryan Catanzaro. 2018. Image inpainting for irregular holes using partial convolutions. In *ECCV*. Springer, USA, 85–100.

[17] Yuxian Meng, Wei Wu, Fei Wang, Xiaoya Li, Ping Nie, Fan Yin, Muyu Li, and Qinghong Han. 2019. Glyce: Glyph-vectors for Chinese character representations. In *NIPS*. Curran Associates, USA, 2746–2757.

[18] Fang Miao and Li Feng. 2020. Research on Character Image Inpainting based on Generative Adversarial Network. In *ICCST*. IEEE Computer Society, USA, 137–140.

[19] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. 2018. Spectral normalization for generative adversarial networks. In *ICLR*. JMLR.org, USA, 1–26.

[20] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. 2016. Context encoders: Feature learning by inpainting. In *CVPR*. IEEE Computer Society, USA, 2536–2544.

[21] Zhi Qiao, Yu Zhou, Dongbao Yang, Yucan Zhou, and Weiping Wang. 2020. SEED: Semantics Enhanced Encoder-Decoder Framework for Scene Text Recognition. In *CVPR*. IEEE Computer Society, USA, 13528–13537.

[22] Alec Radford, Luke Metz, and Soumith Chintala. 2016. Unsupervised representation learning with deep convolutional generative adversarial networks. In *ICLR*. JMLR.org, USA, 1–16.

[23] Jose A. Rodriguez-Serrano, Albert Gordo, and Florent Perronnin. 2015. Label Embedding: A Frugal Baseline for Text Recognition. *IJCV* 113, 3 (2015), 193–207. https://doi.org/10.1007/s11263-014-0793-6

[24] Prasun Roy, Saumik Bhattacharya, Subhankar Ghosh, and Umapada Pal. 2020. STEFANN: scene text editor using font adaptive neural network. In *CVPR*. IEEE Computer Society, USA, 13228–13237.

[25] Kazuma Sasaki, Satoshi Iizuka, and Edgar Simo-Serra. 2017. Joint gap detection and inpainting of line drawings. In *CVPR*. IEEE Computer Society, USA, 5725–5733.

[26] Kazuma Sasaki, Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa. 2018. Learning to restore deteriorated line drawing. *The Visual Computer* 34, 6-8 (2018), 1077–1085.

[27] 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. *PAMI* 39, 11 (2016), 2298–2304.

[28] Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. In *ILSVRC Workshop*. JMLR.org, USA, 1–14.

[29] Tzu-Ray Su and Hung-Yi Lee. 2017. Learning chinese word representations from glyphs of characters. In *EMNLP*. ACL, USA, 1–10.

[30] M Sun, J Li, Z Guo, Z Yu, Y Zheng, X Si, and Z Liu. 2016. Thuctc: an efficient chinese text classifier. *GitHub Repository* (2016).

[31] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NIPS*. Curran Associates, USA, 1–11.

[32] Tomas Wilkinson and Anders Brun. 2016. Semantic and Verbatim Word Spotting Using Deep Neural Networks. In *ICFHR*. IEEE Computer Society, USA, 307–312.

[33] Chao Yang, Xin Lu, Zhe Lin, Eli Shechtman, Oliver Wang, and Hao Li. 2017. High-Resolution Image Inpainting using Multi-Scale Neural Patch Synthesis. In *CVPR*. IEEE Computer Society, USA, 4076–4084.

[34] 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 *CVPR*. IEEE Computer Society, USA, 12113–12122.

[35] Xiang Zhang and Yann LeCun. 2017. Which encoding is the best for text classification in chinese, english, japanese and korean? *arXiv preprint arXiv:1708.02657* (2017), 1–24.

[36] Xinyu Zhou, Cong Yao, He Wen, Yuzhi Wang, Shuchang Zhou, Weiran He, and Jiajun Liang. 2017. East: an efficient and accurate scene text detector. In *CVPR*. IEEE Computer Society, USA, 5551–5560.