Few-shot Font Generation for Japanese Kuzushiji with Differentiable Renderer

Honghui Yuan[1,a)] Junwen Chen[1,b)] Keiji Yanai[1,c)]

Abstract

The study of ancient documents is important for humanity to understand history. In Japan, many ancient documents were written in the Kuzushiji font. Unlike modern fonts, Kuzushiji, as a traditional handwritten font, has unique text structures, with strokes often being connected, simplified, or distorted. In recent years, with the development of deep learning, font image generation has achieved significant success. However, most studies on font generation have focused on modern font generation, and few have focused on the generation of ancient handwritten fonts like Kuzushiji. Furthermore, vector images are resolution-independent, possess high editing flexibility, and have consistent output quality when compared to raster images. Thus, we propose a new few-shot font generation method for generating Kuzushiji characters from modern font based on vector images in this study. We conducted numerous experiments on the generation of Kuzushiji fonts, and the results demonstrate that the proposed method successfully generated Kuzushiji characters in vector images and achieved excellent results.

1. Introduction

Kuzushiji, a font commonly used in ancient Japanese books, recognizes and generates these characters in digital form, which are crucial for the preservation of ancient documents and the understanding of Japanese culture. In recent years, with the development of deep learning, many studies on font generation have achieved excellent results. Most methods based on the Generative Adversarial Network (GAN) and Diffusion model use a few reference images as styles and transform content images into corresponding styles to generate new font images. However, these methods are mainly applicable to modern typography fonts. Font generation for Kuzushiji has several major difficulties compared to that for modern texts. First, the images for the Kuzushiji text are all from ancient documents, leading to an unbalanced dataset and low-quality images. Second, as a handwritten font, Kuzushiji characters have different features from typography fonts and are harder to generate.

[1] The University of Electro-Communications, JAPAN
[a)] yuan-h@mm.inf.uec.ac.jp
[b)] chen-j@mm.inf.uec.ac.jp
[c)] yanai@cs.uec.ac.jp

Fig. 1 The results of our proposed method.

Third, summarizing the Kuzushiji style is challenging, because even the same text from the same author and work has completely different writing methods. Therefore, it is essential to conduct specific research on Kuzushiji fonts. In addition, many current font generation studies are based on pixel-level generation, while there are relatively few studies on vector-based font generation. The vector-based approach can be more effective for the preservation of ancient texts like Kuzushiji characters, which were written in hand. Based on our current knowledge, there has been no research on generating vector-based images of ancient handwritten texts in the existing font generation tasks.

In this study, we proposed a few-shot font generation method to generate Kuzushiji characters using vector-based images without training. Our method requires only one or a few Kuzushiji images as a reference, thus effectively alleviating the problem of unbalanced Kuzushiji databases. We used a differentiable rasterizer and a conditional diffusion model based on Word-As-Image [1]. We also applied the discriminator and the encoder module to convert modern texts to Kuzushiji character fonts. Our method represented the font characters as a set of control points and updated the parameters using several proposed loss functions. As shown in Fig. 1, our method can convert characters in modern font to the corresponding characters with the Kuzushiji style in vector images.

2. Related Work

2.1 Few-shot Font Generation

Few-shot Font Generation (FFG) methods aim to generate a large number of target character fonts in the desired style from only a few reference images. In recent years, various FFG methods have been proposed and have achieved remarkable results. For example, MX-Font [5] was a weakly supervised method, automatically extracting multiple style features by multiple experts without being explicitly conditioned on component labels. DG-Font [13] proposed deformable convolutional blocks in the skip connec-

tion to achieve unsupervised learning. XMPFont [4] proposed a self-supervised cross-modality pre-training strategy and used stroke labels instead of component labels as the basic expression of word structure to achieve high performance in font generation. FSFont [10] realized the few-shot font generation by learning the local style of the reference image and the spatial correspondence between the content and the reference.

Although these methods have achieved great results in various font generation, they are all designed for modern text generation. Furthermore, existing methods always need the TrueType Font(TTF) file for training. However, handwritten characters do not have the TTF file and these methods cannot be used for ancient handwritten fonts like Kuzushiji. Thus, in this study, we aim to achieve font generation of handwritten Kuzushiji characters using the Few-shot Font Generation (FFG) method, which is capable of generating images from a small number of reference images selected from ancient books.

2.2 Methods with differentiable renderers

Different from previous research that generated raster images, many recent studies have shifted font generation tasks to the vector domain. CLIPFont [9] used CLIP [7] to achieve font style transfer by calculating the distance between the prompt and images. Zero-shot-font [2] achieved font style transfer by combining CLIP with the differentiable renderer and distance transfer loss function. Word-As-Image [1] used CLIP and diffusion model to transfer the semantic meaning of the input word into the corresponding character in the word. DS-Fusion [11] could stylize the English letter font, visually convey the semantics of the input word to the letter, and ensure the output results remain readable. VecFusion [12] utilized the raster diffusion model and vector diffusion model in cascade mode and made a vector diffusion model to optimize the font images generated by the raster diffusion model.

The above methods enabled the text generation task to be implemented in the vector domain, however, they are all focused on text artistic style conversion and cannot realize the generation of normal fonts. Furthermore, most of these methods are mainly focused on English character generation and can not apply to complex characters like Japanese. Our study aims to propose a method for generating Japanese Kuzushiji characters in the vector domain without the need to train the model.

3. Methodology

3.1 Basic method Word-As-Image

Firstly, we will introduce the Word-As-Image [1] that is used as our base network, which uses a pre-trained stable diffusion model to transfer the semantic features from input prompt to images. Fig. 2 shows the framework of Word-As-Image. This model uses Bezier Curves to deform the target characters by specifying the style from the input prompt. Since the Stable Diffusion model [6] operates

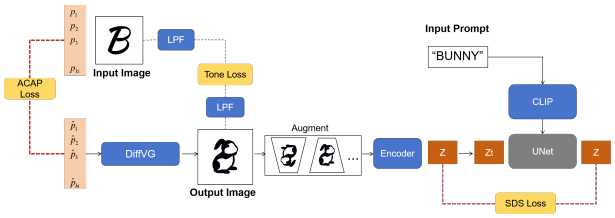

Fig. 2 The framework of Word-As-Image.

on raster images, it uses DiffVG [3] to optimize shape parameters. Word-As-Image applies the semantic features of word to the characters belonging to this word individually to produce a semantic visual depiction of the characters. It utilizes the FreeType font library[*1] to extract the outline of the character. The extracted outline is defined by a different number of control points $P_i = \{P_j\}_{j=1}^{k_i}$ for the target character l_i.

During the generation process, the model incorporates three loss functions to optimize the above parameters. The SDS loss $\nabla_\theta \mathcal{L}_{LSDS}$ could convey semantic concepts from the prompt to the images using CLIP [7] and the Stable Diffusion model [6]. This loss function can be defined as follows:

$$\nabla_\theta \mathcal{L}_{LSDS} = \mathbb{E}_{t,\epsilon}\left[w(t)\left(\hat{\epsilon}_\phi(\alpha_t x_t + \sigma_t \epsilon, y) - \epsilon\right)\frac{\partial z}{\partial z_{aug}}\frac{\partial x_{aug}}{\partial \theta}\right] \tag{1}$$

The As-Conformal-As-Possible Deformation Loss $\mathcal{L}_{acap}$ is designed to preserve the structure of the font. This loss function encourages the optimized shape $\hat{P}$ to have an induced angle α that does not deviate significantly from the angle of the original shape P.

$$\mathcal{L}_{acap}(\mathbf{P}, \hat{\mathbf{P}}) = \frac{1}{k}\sum_{j=1}^{k}\left(\sum_{i=1}^{m_j}(\alpha_j^i - \hat{\alpha}_j^i)^2\right) \tag{2}$$

where α_j^i denotes the angles between the connection line of control points.

The Tone Preserving Loss $\mathcal{L}_{\text{tone}}$ is designed to maintain the consistency of the font's tone (the total amount of black and white areas) between the generated images and the input images.

$$\mathcal{L}_{\text{tone}} = \left\|\text{LPF}(R(P)) - \text{LPF}(R(\hat{P}))\right\|_2^2 \tag{3}$$

where LPF represents a low-pass filter and P and $\hat{P}$ are the sets of control point parameters that constitute the vector image.

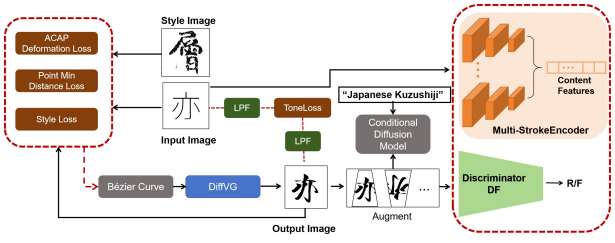

Fig. 3 The network of the proposed method.

[*1] https://freetype.org/

3.2 Proposed method

Word-As-Image achieved the transformation from the semantic features of the prompts into character images of the corresponding words and yielded excellent results. However, this method has several limitations. It mainly focuses on English characters and cannot produce satisfactory results for complex text with many strokes like Japanese characters. Moreover, when presented with sparse prompts like "Japanese Kuzushiji," the model cannot fully understand the semantic meaning of the prompt. This method is also limited to converting semantic styles to the characters that belong to the input semantic words, which does not allow for the free selection of characters.

Therefore, we propose a new model to generate vector images for Kuzushiji characters using a few reference images without training. We use "Japanese Kuzushiji" as the input prompt and our method can select any text to be converted without the limitation of the generation in the corresponding words. The framework of the proposed method is illustrated in Fig. 3. When using the prompt as a condition to control the generation of fonts, it is challenging to describe Kuzushiji features in terms of the prompt. Therefore, we use the text image of Kuzushiji to control the generated result. In addition to using the Conditional Stable Diffusion model [6] and DiffVG used by Word-As-Image, we utilize a Discriminator(DF) that is trained to discriminate Kuzushiji character images and modern font images. Specifically, we utilize the Kuzushiji dataset and modern text to train the discriminator in advance. In the generation process, we utilize the trained discriminator to control the generation of Kuzushiji characters. In addition, inspired by the MX-Font [5], we applied the Multi-Stroke encoders to control the generation at the detail level. Furthermore, we proposed several new loss functions to realize font transfer during the few-shot Kuzushiji font generation process.

3.2.1 Style Loss

To better achieve the style transfer of the Kuzushiji features for the generated characters, we used the style loss L_{style} in our method. Specifically, we compute the Gram matrix for both the features of the generated images and the style images. After calculating these matrices, we compute the mean squared error between them. The style loss allows us to quantitatively measure and minimize the stylistic differences between the two images. The style images were randomly selected from the Kuzushiji dataset, and to better obtain style features and reduce the influence of content features, in this study, we used 20 style images to extract the mean style features. The calculation of the style loss is as follows.

$$L_{\text{style}} = \sum \|G_{l,i,j}(x_s) - G_{l,i,j}(x_g)\|^2 \tag{4}$$

$$G_l(x) = F_l(x)F_l^T(x) \tag{5}$$

where G_l is the Gram matrix. $F_l(x)$ is the feature map of image x from the VGG19 [8]. x_s is the reference style image and x_g is the generated image.

3.2.2 Multi-stroke ACAP Loss

In Word-As-Image [1], the As-Conformal-As-Possible Deformation Loss function helps control the angle between control points, preventing large deformations. To more precisely control the font structure of complex characters that have many strokes, the angle calculation for the entire character is necessary. The original loss function performs very well for simple English letters, but due to the multi-stroke structure of Kuzushiji characters, it is sometimes only calculating small parts of the text and impossible to calculate the angle for the entire character. Thus, it does not perform well in complex font characters. In this study, we proposed Multi-stroke ACAP Loss L_{ACAP} to extend this loss function to apply to complex font structure that has multi-strokes. Specifically, instead of building control points for the whole character at once, we use Diffvg to get the number of strokes in the font and then construct control points for each stroke in the characters and combine them to get the final control points. The visualization results are shown in the supplementary materials.

3.2.3 Point Min Distance Loss

In addition to constraints on the overall appearance of the font and the angle between the control points, we also utilized the minimum distance loss L_{point} to restrict the length between control points. We prevent sudden and significant changes in the connecting lines between control points to ensure that the distances of the characters change gradually, thereby maintaining the continuity of the strokes in the character. Specifically, we calculated the distance between adjacent points of the control point and controlled the distance within a certain value.

3.2.4 Multi-Stroke Module

Although we employ the aforementioned loss function to control style and font structure, they are all grounded in the overall structure of the font and lack component-level control. Furthermore, because kuzushiji is often accompanied by large deformation of font structure, without detailed control, it is easy to cause the generated results to be accompanied by the collapse of structure. We will prove this in ablation studies. Therefore, to add the detailed features, we added the Multi-Stroke Module to our network and calculate the loss function L_{SLS} to ensure the readability and integrity of the font. Specifically, as shown in Fig. 3, drawing inspiration from MX-Font [5] that uses multiple encoders to automatically extract the component features of characters, we introduced the multiple-stroke encoder to extract stroke-level features. Specifically, we use 4 encoders to extract the detailed features and then concatenate these features. After getting the whole feature, we calculate the Mean Squared Error (MSE) loss between the generated image and the original content image.

Additionally, we used Tone Preserving Loss L_{tone} to preserve the adjusted image without deviating significantly from the input image. And we also used the SDS loss to transfer the semantic features of the prompt into the latent space. The total loss function of our method is as follows:

$$L_{total} = \lambda_{dis} L_{\text{discriminator}} + \lambda_t L_{\text{tone}} + \lambda_a L_{\text{ACAP}} + \lambda_p L_{\text{point}} + \lambda_s L_{\text{style}} + \lambda_{sls} L_{\text{SLS}} + \lambda_{sds} L_{\text{SDS}} \quad (6)$$

By utilizing the loss functions and module described above, our model can precisely manipulate various aspects of the vector image, including controlling the angles between control points, the distances between these points, the overall structure and detailed strokes of the font. Thus, we can transfer the Kuzushiji style features to the modern font.

4. Experiments

4.1 Experimental Settings and Results

Our experiments were conducted with the following settings. The number of Iterations: 300. Point Min Distance Loss Distance: 1. The loss weights: λ_{dis}: 0.01 , λ_t: 1, λ_{sds}: 0.001, λ_a: 1, λ_p: 1, λ_s: 0.002, λ_{sls}: 1. We used version 1.5 of the Stable Diffusion model. The input in the reference phase only needs the specified characters. The reference time of our method is about 2 to 3 minutes. The experiments are conducted on a single RTX 3090 GPU.

As shown in Fig. 1, the experimental results of the proposed method show that our method can realize Kuzushiji character generation from source modern font characters while ensuring the readability of the text. Compared to the real Kuzushiji characters, our results could generate the same features at the stroke level.

4.2 Comparison with Previous Method

In this section, we will compare the results of our method with those of previous methods. The objective of our experiment is to generate Kuzushiji vector images from modern type font images. Word-As-Image and CLIPFont are methods that, like ours, utilize vector images to transfer font style based on input prompt. Thus, we conducted experiments between our method and these methods. In both experiments, Kuzushiji characters were not learned in advance. The input prompt for the other methods was the same as ours, which is "Japanese Kuzushiji". Furthermore, we also compare with recent raster-based font generation methods that utilize Kuzushiji images as style references. Fig. 4 shows the comparison results. The results from CLIPFont are similar to the original font with few Kuzushiji features. Word-As-Image exhibits only irregular deformation of characters without distinct Kuzushiji features. Other font generation methods like DG-Font only slightly reflect the Kuzushiji style. In contrast, our method successfully generates Kuzushiji characters from the source modern font characters and generates cleaner and more structurally accurate glyphs compared to the real Kuzushiji font.

Fig. 4 Comparison results between our method and previous methods.

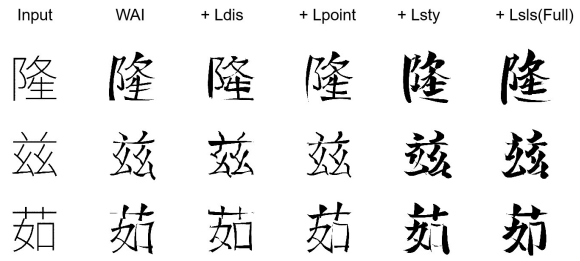

Fig. 5 The results of ablation study.

4.3 Ablation Studies

We conducted ablation studies to verify the effectiveness of each part of the proposed method. The results of the ablation study are shown in Fig. 5. We only investigated the new loss functions and modules that were proposed in this study. WAI represents the results generated using SDS, basic ACAP, and Tone loss in the basic method Word-As-Image. When we incorporate the discriminator into the model, the generated results have some features of Kuzushiji, but the features are not uniformly distributed among the strokes and the effect is not obvious. By utilizing Point Min Distance Loss, the stylistic characteristics of Kuzushiji can be rendered more apparent. The results from utilizing style loss indicate that style loss effectively transfers the style of the Kuzushiji style to the whole image but causes distortions that appear at the stroke level. Especially unnecessary deformations at the ends of strokes. As shown in the last column, after adding the Multi-Stroke module, the strokes of the characters are successfully preserved in the generated results while maintaining the feature of Kuzushiji. Therefore, by combining the aforementioned loss functions and modules, our method can generate natural-looking Kuzushiji characters while ensuring the complete structural integrity of the font.

5. Conclusion

In this study, we proposed a few-shot font generation model for generating Kuzushiji characters in vector images without training. The proposed method can generate Kuzushiji character fonts from modern fonts according to the reference Kuzushiji characters. Our experimental results demonstrate that this approach can maintain the structural integrity of the text while accurately reflecting the Kuzushiji style in the detail level. We have analyzed the effectiveness of each loss function and module in our method. The results show that with all of the advantages of our component, we could generate characters that captured the Kuzushiji features and generate the closest results to the real Kuzushiji image. Considering that ancient texts often suffer from a lack of sufficient and high-quality datasets, our method provides a convenient means for generating Kuzushiji characters without the need for a large amount of text data and thus can contribute to enriching the dataset as well as recovering ancient texts in other languages. Furthermore, because our model only needs to input text, it can be easily implemented for large-scale generation tasks. Future works such as texture generation and precise generation need to be solved.

References

[1] Iluz, S., Vinker, Y., Hertz, A., Berio, D., Cohen-Or, D. and Shamir, A.: Word-as-image for semantic typography, *ACM Transactions on Graphics (TOG)*, Vol. 42, No. 4, pp. 1–11 (2023).

[2] Izumi, K. and Yanai, K.: Zero-shot font style transfer with a differentiable renderer, *Proceedings of the 4th ACM International Conference on Multimedia in Asia*, pp. 1–5 (2022).

[3] Li, T.-M., Lukáč, M., Gharbi, M. and Ragan-Kelley, J.: Differentiable vector graphics rasterization for editing and learning, *ACM Transactions on Graphics (TOG)*, Vol. 39, No. 6, pp. 1–15 (2020).

[4] Liu, W., Liu, F., Ding, F., He, Q. and Yi, Z.: Xmp-font: Self-supervised cross-modality pre-training for few-shot font generation, *Proc. of IEEE Computer Vision and Pattern Recognition*, pp. 7905–7914 (2022).

[5] Park, S., Chun, S., Cha, J., Lee, B. and Shim, H.: Multiple heads are better than one: Few-shot font generation with multiple localized experts, *Proc. of IEEE International Conference on Computer Vision*, pp. 13900–13909 (2021).

[6] Poole, B., Jain, A., Barron, J. T. and Mildenhall, B.: DreamFusion: Text-to-3D using 2D Diffusion, *International Conference on Learning Representations* (2023).

[7] Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J. et al.: Learning transferable visual models from natural language supervision, *Proc. of the International Conference on Machine Learning*, PMLR, pp. 8748–8763 (2021).

[8] Simonyan, K. and Zisserman, A.: Very deep convolutional networks for large-scale image recognition, *arXiv preprint arXiv:1409.1556* (2014).

[9] Song, Y. and Zhang, Y.: CLIPFont: Text Guided Vector WordArt Generation, *Proc. of the British Machine Vision Conference*, BMVA Press (2022).

[10] Tang, L., Cai, Y., Liu, J., Hong, Z., Gong, M., Fan, M., Han, J., Liu, J., Ding, E. and Wang, J.: Few-shot font generation by learning fine-grained local styles, *Proc. of IEEE Computer Vision and Pattern Recognition*, pp. 7895–7904 (2022).

[11] Tanveer, M., Wang, Y., Mahdavi-Amiri, A. and Zhang, H.: Ds-fusion: Artistic typography via discriminated and stylized diffusion, *Proc. of IEEE International Conference on Computer Vision*, pp. 374–384 (2023).

[12] Thamizharasan, V., Liu, D., Agarwal, S., Fisher, M., Gharbi, M., Wang, O., Jacobson, A. and Kalogerakis, E.: VecFusion: Vector Font Generation with Diffusion, *Proc. of IEEE Computer Vision and Pattern Recognition*, pp. 7943–7952 (2024).

[13] Xie, Y., Chen, X., Sun, L. and Lu, Y.: Dg-font: Deformable generative networks for unsupervised font generation, *Proc. of IEEE Computer Vision and Pattern Recognition*, pp. 5130–5140 (2021).