KuzushijiFontDiff: Diffusion Model for Japanese Kuzushiji Font Generation

Honghui Yuan[0009−0001−4334−9363] and Keiji Yanai[0000−0002−0431−183X]

The University of Electro-Communications, Chofu, Tokyo, JAPAN
{yuan-h, yanai}@mm.inf.uec.ac.jp

Fig. 1. The Kuzushiji font results generated by our application.

Abstract. Kuzushiji is an ancient type of font that was used in Japan hundreds of years ago, and many valuable historical documents were written in this font. These documents are crucial for understanding Japan's past. Kuzushiji differs from modern fonts due to its complex structure, large deformations, and continuous strokes. However, existing font generation methods have mainly focused on modern fonts, and few studies have targeted the generation of ancient fonts like Kuzushiji. Current methods do not perform well in generating Kuzushiji characters. Therefore, to solve this problem, we have proposed a web-based real-time application for generating Kuzushiji fonts. This application allows users to freely select Japanese characters and generate corresponding Kuzushiji character images. As shown in Fig.1, our application can convert modern text into Kuzushiji text efficiently. This application is based on a conditional diffusion model, incorporating a stroke-level style module and a content-retaining module, which we have proposed.

Keywords: Japanese Kuzushiji · font generation · diffusion model

1 Introduction

Designing fonts is a time-consuming and labor-intensive task, requiring specialized knowledge, especially for fonts with complex structures such as Chinese and

Japanese. In recent years, deep learning networks have made significant progress in font generation. Many methods can now generate a target font image in the style of a reference font by combining the content features of the target characters with the style features of the reference font image. With the development of models like Transformer [17] and Diffusion [12], high-quality font generation for a large number of fonts has also achieved promising results.

The Kuzushiji font, a Japanese script used before the Edo period, is crucial for understanding Japan's history. However, as time passes, more historical documents are being destroyed, and fewer books are being digitized, making the preservation and digitization of Kuzushiji an important area of study. The existing Kuzushiji data faces several challenges. First, the dataset is highly imbalanced. The text with the most images contains 41,293 instances, while the majority of categories have very few images. 790 categories have only one image each. Second, high-quality images of Kuzushiji characters are difficult to obtain. Many of the images in the dataset are directly taken from ancient books, resulting in low-resolution images with significant noise. This highlights the need for a fast and efficient Kuzushiji font generation application. While recent approaches using diffusion models have achieved good results in generating a large number of fonts, including those with complex structures like Chinese and Japanese, these studies are primarily focused on modern fonts. Kuzushiji, as an ancient script, differs from modern fonts in that it has unique characteristics, such as continuous strokes and more significant deformations. Existing methods have not performed well in generating Kuzushiji fonts, often failing to capture the Kuzushiji style or producing unrecognizable structures. Given these challenges, research specifically focused on Kuzushiji font generation is necessary. Furthermore, there is a need for an application capable of generating Kuzushiji text quickly and efficiently.

This study focuses on the generation of Kuzushiji fonts in real-time. We designed a Kuzushiji font generation network based on a conditional diffusion model that solved the problem of generation of Kuzushiji font, and a simple and efficient web-based editor to generate Kuzuhshiji font in real-time. Using our application can help solve the problem of digitizing Kuzushiji.

2 Related Work

2.1 Font Generation

The development of font generation models has achieved promising results. In recent years, several font generation methods have been introduced based on GAN(Generative Adversarial Networks) [4]. LF-Font [9] decomposed characters into components and proposed learning localized component-wise style representations. DM-Font [2] proposed a dual memory architecture to generate the font. MX-Font [10] divided the fonts into several components, and the stylistic features of each component were automatically extracted by several experts. CG-GAN [7] separated content and style features in each component through component discriminators and optimized the font generators. DG-Font [19] achieved unsuper-

vised learning by introducing a deformation skip connection to learn transformation between different fonts. CF-Font [18] further extended DG-Font by using the content fusion module to obtain a robust content feature. Furthermore, XMP-Font [8] proposed a self-supervised cross-modality pre-training strategy and an encoder with a cross-modality transformer, it requires only one reference glyph and achieves the style transfer with high performance. FSFont [14] used cross-attention of reference and content to generate the local structure of the font. However, these methods struggle with issues such as incomplete font strokes and distortion of the font, particularly in fonts with complex structures, such as Chinese. As a result, they have not performed well in generating fonts with complex designs and numerous strokes, such as the Kuzushiji font.

2.2 Differentiable renderers methods

In recent years, methods that optimize the parameters of Bessel curves using differentiable renderers have been applied to font generation tasks. Building on the development of the CLIP [11], which used 400 million image-text pairs to learn the similarity between various images and texts, CLIPFont [13] achieved zero-shot font generation. By associating font generation with style transformation through natural language, CLIPFont enabled the generation of specific font styles from textual descriptions. Zero-shot-font [6] successfully implemented prompt-based text style transformation by utilizing Distance Transform Loss [1], the CLIP model, and the differentiable renderer. DS-Fusion [15] utilized style prompts and text images as input. The style images are generated based on the style words, and then the style features are fused into the font using the latent diffusion process by Latent Diffusion Model (LDM) [12] and a discriminator. Word-As-Image [5] achieved the stylistic transformation of text based on the meaning of input words using a pre-trained stable diffusion model. It maintained the readability of the text while providing a visual representation of the corresponding words. Although these Bezier curve-based font generation methods have produced high-quality font images, they primarily focus on artistic font generation and cannot convert and generate common fonts.

2.3 Diffusion model methods

Recently, several approaches have utilized style images as conditional inputs into diffusion models to generate images consistent with the style of the condition. In the task of font generation, many methods utilized conditional diffusion models to achieve font transfer based on the reference font images, and great results have been achieved. FontDiffuser [20] used multi-scale content features and a style contrastive learning strategy to generate the fonts based on the reference images. FontDiffuser performed well in generating complex characters like Chinese characters with many strokes and can handle larger style variations than previous methods. Generate Like Experts [3] used the diffusion model and divided font generation into a multi-stage process. This method incorporated the source and target fonts in the intermediate stages of the noise addition and

denoising process of the diffusion model. This method aligned the distribution during the font transfer process and achieved high-quality font generation with just a few reference samples. Furthermore, VecFusion [16] combined the diffusion model and vector module in the font generation task. The vector module was used as the super-resolution component. Like the aforementioned methods, our application is based on a diffusion model and uses reference images to generate fonts. However, previous methods have focused on modern fonts, they do not perform well in generating ancient fonts like Kuzushiji. Our application is specifically designed for generating ancient Kuzushiji fonts.

3 Proposed System

3.1 System Overview

We developed a web-based application for real-time Kuzushiji font generation. The input of our application is a content image and a Kuzushiji reference image, and the output is the Kuzushiji text that maintains text consistency with the content image. The user interface of the system is shown in Fig. 2. When selecting content text, users can choose the content character from the example list displayed at the bottom of the interface or enter text in the input field. The input of content images could also accept the uploaded modern text images from the user. The input of the reference image can also be chosen from the Kuzushiji images we provide in the example list. When the input is prepared, press the "Run Kuzushiji Generation" button to generate the Kuzushiji text image based on the content text and reference image selected by the user. The generated result will then be displayed in the result column. Our application takes only 2 to 3 seconds to generate the corresponding Kuzushiji text.

3.2 Network of Application

To achieve the application for the generation of Kuzushiji font, we proposed a Kuzushiji font generation model to solve the problem of existing methods not being able to generate Kuzushiji fonts efficiently. Our network is based on the FontDiffuser, which is a font generation method using the conditional diffusion model. FontDiffuser has achieved outstanding results in generating over 100 fonts and has achieved state-of-the-art performance, particularly for complex fonts and styles that involve significant variations compared to earlier methods. However, when generating Kuzushiji fonts with unique structures and continuous strokes that are different from modern fonts, FontDiffuser does not perform well because the network does not have enough details features on the stroke level.

The overview of our proposed model is shown in Fig. 3. Specifically, the model is divided into three sub-networks, which are the style model, the content model, and the conditional diffusion model. The diffusion model is used to generate the font images, and the style and content models are used as conditions to control the style features and content features during the generation process. Specifically, based on the network of FontDiffuser, we applied a new multiple heads stroke encoder and the MLP module in the style model to enable our model

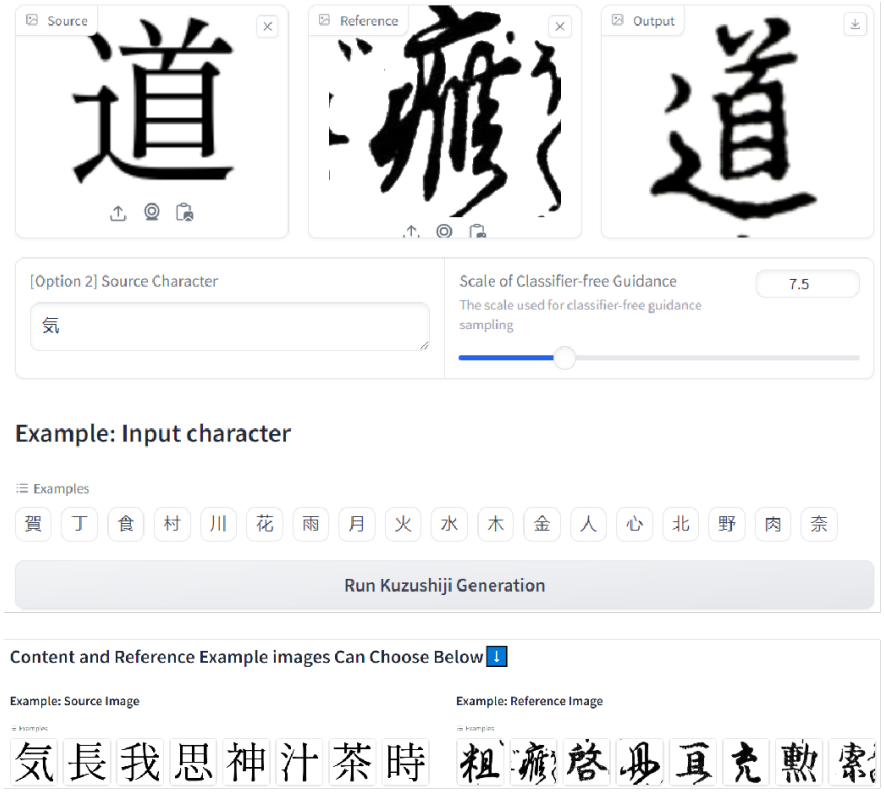

Fig. 2. The interactive interface of our application.

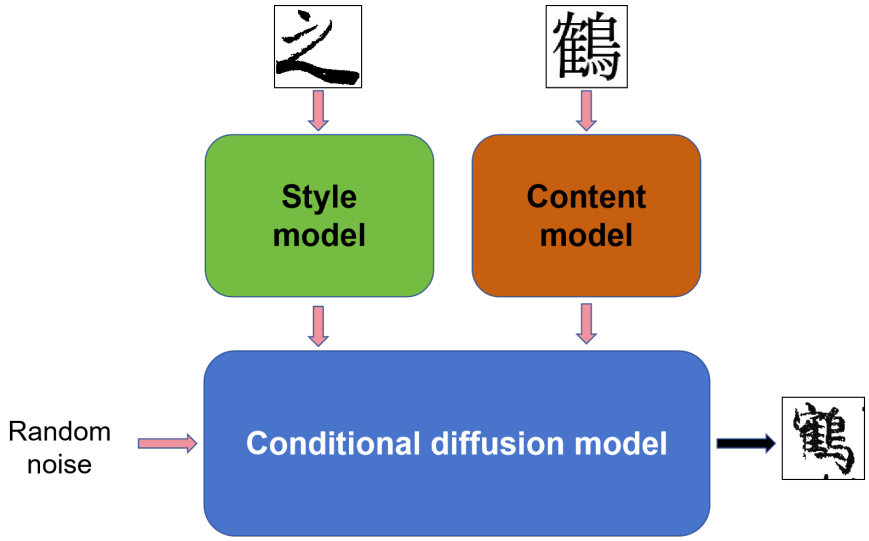

Fig. 3. The network overview of our application.

to achieve Kuzushiji-style transfer at the stroke level. We also added a patch content discriminator in the content model to ensure the stability and readability of the font structure. The results of our proposed method and FontDiffuser are shown in Fig. 4. Compared to our results, the results of FontDiffuser can only generate modern text even used Kuzushiji as the reference image. Our method successfully generated the Kuzushiji font from the modern text and achieved state-of-the-art results in Kuzushiji font generation.

4 Experiments

We previously trained our network using the Kuzushiji dataset, and when the application is running, we utilize the trained model to generate results. Details of the training network are given below. Because the Kuzushiji data is taken from ancient books, many of the data images have background colors, and to focus on text generation, we used black-and-white text images from the dataset as our training data. Specifically, we collected about 4,300 Kuzushiji font images from "Oragaharu" to train our network. When training the network, the batch size of our model was set to 20 and the total step was 62000. The learning rate was 1e–4 and we utilized an RTX4090 for training. We use AdamW optimizer with $\beta 1 = 0.9$ and $\beta 2 = 0.999$. The image size is set as 96×96. We use the DPM-Solver++ sampler with 50 inference steps. Our inference time takes about 2 to 3 seconds.

Fig. 4. The results of our application and baseline FontDiffuser.

5 Conclusion

In this study, we proposed a web-based application for Kuzushiji font generation. This application allows users to transform the modern text into corresponding Kuzushiji text in real-time, helping to address the digitization challenges of Kuzushiji font characters. From the experimental results, we found that while our application successfully generated Kuzushiji fonts for most texts, overly complex fonts sometimes resulted in stroke entanglement, reducing readability. In the future, we plan to add a new content-retaining module to solve this problem.

References

1. Atarsaikhan, G., Iwana, B.K., Uchida, S.: Contained neural style transfer for decorated logo generation. In: 2018 13th IAPR International Workshop on Document Analysis Systems (DAS). pp. 317–322 (2018)
2. Cha, J., Chun, S., Lee, G., Lee, B., Kim, S., Lee, H.: Few-shot compositional font generation with dual memory. In: Proc. of European Conference on Computer Vision. pp. 735–751. Springer (2020)
3. Fu, B., Yu, F., Liu, A., Wang, Z., Wen, J., He, J., Qiao, Y.: Generate like experts: Multi-stage font generation by incorporating font transfer process into diffusion models. In: Proc. of IEEE Computer Vision and Pattern Recognition. pp. 6892–6901 (2024)
4. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. Advances in Neural Information Processing Systems **27** (2014)
5. Iluz, S., Vinker, Y., Hertz, A., Berio, D., Cohen-Or, D., Shamir, A.: Word-as-image for semantic typography. ACM Transactions on Graphics (TOG) **42**(4), 1–11 (2023)
6. Izumi, K., Yanai, K.: Zero-shot font style transfer with a differentiable renderer. In: Proceedings of the 4th ACM International Conference on Multimedia in Asia. pp. 1–5 (2022)
7. Kong, Y., Luo, C., Ma, W., Zhu, Q., Zhu, S., Yuan, N., Jin, L.: Look closer to supervise better: One-shot font generation via component-based discriminator. In: Proc. of IEEE Computer Vision and Pattern Recognition. pp. 13482–13491 (2022)
8. Liu, W., Liu, F., Ding, F., He, Q., Yi, Z.: Xmp-font: Self-supervised cross-modality pre-training for few-shot font generation. In: Proc. of IEEE Computer Vision and Pattern Recognition. pp. 7905–7914 (2022)
9. Park, S., Chun, S., Cha, J., Lee, B., Shim, H.: Few-shot font generation with localized style representations and factorization. In: Proc. of the AAAI Conference on Artificial Intelligence. vol. 35, pp. 2393–2402 (2021)
10. Park, S., Chun, S., Cha, J., Lee, B., Shim, H.: Multiple heads are better than one: Few-shot font generation with multiple localized experts. In: Proc. of IEEE International Conference on Computer Vision. pp. 13900–13909 (2021)
11. 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. In: International Conference on Machine Learning. pp. 8748–8763 (2021)
12. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: Proc. of IEEE Computer Vision and Pattern Recognition. pp. 10684–10695 (2022)
13. Song, Y., Zhang, Y.: Clipfont: Text guided vector wordart generation. In: 33rd British Machine Vision Conference 2022, BMVC 2022, London, UK, November 21-24, 2022. BMVA Press (2022), https://bmvc2022.mpi-inf.mpg.de/0543.pdf
14. Tang, L., Cai, Y., Liu, J., Hong, Z., Gong, M., Fan, M., Han, J., Liu, J., Ding, E., Wang, J.: Few-shot font generation by learning fine-grained local styles. In: Proc. of IEEE Computer Vision and Pattern Recognition. pp. 7895–7904 (2022)
15. Tanveer, M., Wang, Y., Mahdavi-Amiri, A., Zhang, H.: Ds-fusion: Artistic typography via discriminated and stylized diffusion. In: Proc. of IEEE International Conference on Computer Vision. pp. 374–384 (2023)

16. Thamizharasan, V., Liu, D., Agarwal, S., Fisher, M., Gharbi, M., Wang, O., Jacobson, A., Kalogerakis, E.: Vecfusion: Vector font generation with diffusion. In: Proc. of IEEE Computer Vision and Pattern Recognition. pp. 7943–7952 (2024)
17. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Advances in Neural Information Processing Systems **30** (2017)
18. Wang, C., Zhou, M., Ge, T., Jiang, Y., Bao, H., Xu, W.: Cf-font: Content fusion for few-shot font generation. In: Proc. of IEEE Computer Vision and Pattern Recognition. pp. 1858–1867 (2023)
19. Xie, Y., Chen, X., Sun, L., Lu, Y.: Dg-font: Deformable generative networks for unsupervised font generation. In: Proc. of IEEE Computer Vision and Pattern Recognition. pp. 5130–5140 (2021)
20. Yang, Z., Peng, D., Kong, Y., Zhang, Y., Yao, C., Jin, L.: Fontdiffuser: One-shot font generation via denoising diffusion with multi-scale content aggregation and style contrastive learning. In: Proc. of the AAAI Conference on Artificial Intelligence. vol. 38, pp. 6603–6611 (2024)