KuzushijiDiffuser: Japanese Kuzushiji Font Generation with FontDiffuser

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

Abstract. Kuzushiji characters were used in Japan hundreds of years ago, and many valuable ancient documents are written in Kuzushiji. Research into generating Kuzushiji characters increases the training data for recognizing these characters and enhances people's interest and knowledge of Kuzushiji, which is crucial for understanding Japan's past. However, few studies have focused on the Kuzushiji font generation task. A significant challenge in studying Kuzushiji font generation is the increasing number of deteriorated historical documents, leading to some characters being lost or illegible. In recent years, image generation using deep learning has made remarkable progress, and various methods for generating fonts have been developed with great success. However, because Kuzushiji is different from ordinary fonts, with complex structures that have larger deformations and continuous strokes, existing font generation methods do not perform well in generating Kuzushiji font. To address this problem, we propose a Kuzushiji font generation framework based on the diffusion model. Specifically, based on FontDiffuser, we introduce a stroke style module to efficiently extract font style at the stroke level, and we utilize a content-retaining model to ensure the structural integrity of the font. We conducted extensive experiments on a large number of fonts, and the results show that our model is capable of generating high-quality Kuzushiji characters, achieving state-of-the-art performance.

Keywords: Kuzushiji characters · font generation · FontDiffuser

1 Introduction

Kuzushiji characters are Japanese characters written before the Edo period. As fewer historical documents are digitized over time, Kuzushiji characters are studied less frequently, making it difficult for non-specialists to recognize classical Japanese characters. Although research is underway on the reprinting and digitization of historical documents, the deterioration of these documents increases with time, leaving many literary works still buried in the past without being digitized. The scarcity and illegibility of data have made the task of the Kuzushiji font generation challenging. Recently, various efforts have been made to improve the digitization of literary works by using deep learning to recognize Kuzushiji

characters. In the Kaggle competition held in 2019, Kuzushiji character recognition achieved over 90 percent accuracy for Edo period Kuzushiji characters, a practical level of recognition for digitizing these characters. However, few studies have focused on the Kuzushiji font generation. The font generation task involves creating characters for a new font by converting an image from the source domain to the target domain based on the reference image. Font generation has wide applications in refining fonts for ancient books and designing new fonts. In recent years, various studies have explored font generation using image generation models, enabling the creation of various fonts with high performance. However, these studies are limited to modern character generation, and even when retraining networks with Kuzushiji data, the performance is unsatisfactory. Since Kuzushiji characters exhibit different features, such as more deformation in structure compared to modern fonts and continuous stroke structure, research specifically focused on Kuzushiji characters is necessary.

In this study, we propose "KuzushijiDiffuser," a Kuzushiji font generation model based on the diffusion model. Our approach introduces a new stroke-level stylistic feature extractor that can capture detailed stylistic features. This module enables our model to achieve Kuzushiji-style transfer at the stroke level. Additionally, since Kuzushiji characters exhibit greater deform and ligatures compared to modern fonts, maintaining the structure and readability of the font can be more challenging. To ensure the stability and readability of the font structure, we introduce a content-retaining model to enhance the model's ability to remain invariant to the font structure. To verify the effectiveness of our model for Kuzushiji font generation, we conducted experiments on a large set of characters, including a variety of kanji with different strokes. The experimental results demonstrate that our model performs well in generating Kuzushiji font.

Our contribution is summarized as follows.

1. We propose a Kuzushiji generation framework that can effectively generate Japanese characters in the Kuzushiji style.
2. Our method uses any single Kuzushiji character image as the reference to generate Kuzushiji characters.
3. The experiments have demonstrated that our method can generate visually appealing Kuzushiji characters and achieve state-of-the-art results.

2 Related Work

2.1 Font Generation

In recent years, various font generation methods have been proposed to create target character fonts in a desired style from reference images. These methods have the potential to greatly reduce the time-consuming and labor-intensive process involved in designing fonts for languages with a large number of characters.

Some font generation methods were regarded as tasks of Image-to-Image Translation. For example, Rewrite[1] was based on the pix2pix [13] framework and

[1] Rewrite. https://github.com/kaonashi-tyc/rewrite.

learned to transfer font style. However, the network learned to map the source font style to the specific target font style, so it needed to be relearned for the new font style. Similarly, zi2zi [4] learned multiple font styles by incorporating an embedding of predefined style categories, but it could not generalize to an unknown font style not included in the predefined styles. Subsequently, EMD [31] and SA-VAE [25] were proposed to separate style and content features, allowing generalization to new styles. However, these models sometimes failed to capture local style patterns, leading to undesired results.

Moreover, Image-to-Image Translation methods like UNIT [16] extended CoGAN [18], an adversarial generative network (GAN) [8], and a variational autoencoder [15], which together to learn the distribution between the domains. MUNIT [9], which realized unsupervised multi-style transfer, is an extension of UNIT [16]. In MUNIT, the content space of the image is assumed to be shared and the style space is assumed to be independent. The multi-domain Image-to-Image Translation was also realized in many studies [5, 6, 1, 11]. Furthermore, FUNIT [17] further extended the generalization capability to unknown domains by learning to encode content and class images with unsupervised data respectively. By training a wide variety of class images in advance, FUNIT [17] achieved high-performance generation of unknown classes, such as font classes, from a small amount of data.

Several component-based methods were proposed like RD-GAN [10]. RD-GAN introduced a Radical Extraction Module (REM) that allows one-shot generation of the unseen glyph but only transfers them to the specific target font style. Advanced architectures such as DM-Font [3] and LF-Font [20] used structure-aware style representations and propose learning localized component-wise style representations. DM-Font [3] introduced a dual memory architecture, and LF-Font [20] used a factorization strategy. All these studies employed supervised learning, requiring paired training data. MX-Font [21] was weak supervision, automatically extracting multiple style features by multiple experts without being explicitly conditioned on component labels. It can extract style features and capture a variety of local concepts. DG-Font [29] achieved unsupervised learning by introducing a deformation skip connection. Furthermore, in recent years, several methods have been proposed to extend the above work. XMPFont [19] 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 [26] successfully reproduced local structure by cross-attention of reference (font: used as key and value) and content (character: used as a query). However, these methods did not work well for generating fonts with complex structures like characters with many strokes, and cannot effectively deal with more extensive structure changes in fonts like Kuzushiji font.

2.2 Methods with differentiable renderers

In recent years, several font generation methods have been proposed that use the differentiable renderer to optimize the parameters of Bezier Curves. Unlike

existing font generation methods, CLIPFont [24] proposed a method with zero-shot font generation for any language based on CLIP [22]. CLIP was a model for learning the relationship between text and images, it is based on 400 million image-text pairs and can compute the similarity between various images and texts. CLIPFont minimized the directional distance between text descriptions and fonts in the CLIP embedding space, resulting in artistic font generation. Zero-shot-font [14] eliminated the need for style image input by combining CLIP and the differentiable renderer. It also used Distance Transform Loss introduced by Atarsaikhan *et al.* [2] to successfully generate text images with styles that match the input prompt while preserving the text shape. DS-Fusion [27] was an artistic typography method that designs character fonts to visually convey the meaning of a word and can generate artistically designed characters. It also employed a Latent Diffusion Model (LDM) [23] denoising generator and added a CNN-based discriminator to match the input style to the input text. Word-As-Image [12] was a semantic typography method that designs characters according to the meaning of the input word. It aimed to convey the meaning of the input word to the black-and-white text without changing the color or texture of the text by using the pre-trained Stable Diffusion model. Although it is possible to generate diverse styles of font using Bezier Curves, these methods mainly focus on artistic font generation and leak the ability to convert between general fonts.

2.3 Methods with Diffusion model

With many conditional diffusion models using style images as conditions to control the generation process, diffusion models have shown promising results in image generation. In font generation tasks, some approaches utilized conditional diffusion models to generate fonts based on reference font images. For example, FontDiffuser [30] leveraged multi-scale content features and introduced an innovative style contrastive learning strategy to generate multiple fonts. Generate Like Experts [7] divided font generation into a multi-stage process within a diffusion model, producing high-quality font images with a small number of reference samples. This method incorporated both content and style fonts at an intermediate stage in the noise addition and denoising process of the diffusion model, using the features of style and content fonts as conditions to generate multiple fonts. Similarly, VecFusion [28] employed the diffusion model, adding a vector module as a super-resolution component to achieve font generation. While our method also utilizes the diffusion model and reference font images for font generation, it differs in that existing methods focus on modern fonts and our method can generate ancient fonts like the Kuzushiji font.

3 Methodology

3.1 Basic Network

Recently, many studies on font generation have achieved significant results. FontDiffuser [30] has demonstrated impressive results in generating fonts across vari-

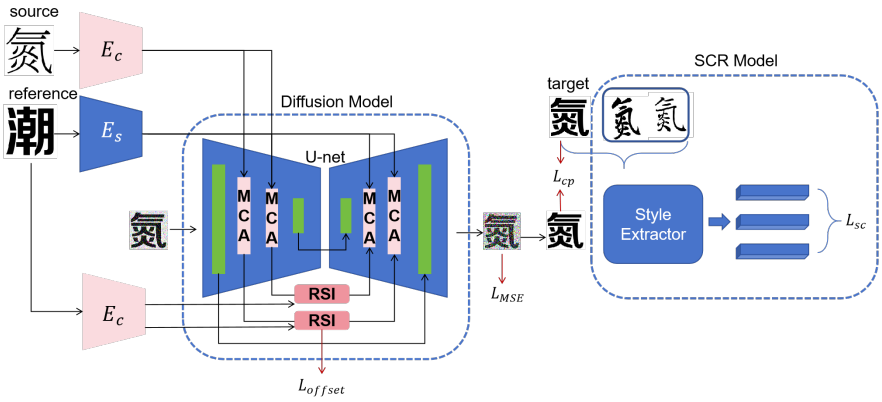

Fig. 1. The overall framework of basic network FontDiffuser.

ous styles. This method is a diffusion-based image-to-image one-shot font generation approach. It has achieved state-of-the-art results, especially with complex fonts and styles involving more significant changes compared to previous methods. Given that Kuzushiji exhibits greater deformation than modern fonts, we used FontDiffuser as our base network for generating Kuzushiji fonts.

Fig. 1 shows the framework of FontDiffuser, which is a font generation method based on the conditional diffusion model. This method comprises four main components: the content encoder E_c, style encoder E_s, conditional diffusion model, and style contrastive refinement (SCR) model. Within the U-net network of diffusion model, a Multi-Scale Content Aggregation (MCA) module and Reference-Structure Interaction (RSI) module are introduced. MCA leverages content and style features from different layers of encoders as multiple scale inputs to compute each cross-attention, ensuring both fine details and overall layout are preserved. RSI module, which incorporates a deformable convolutional network (DCN) and cross-attention mechanism, is used in the skip connections of the U-net to align the structural features of the reference image. Additionally, the training process is divided into two phases. In phase 2, the SCR model used a style extractor to capture the target font and other font style features, with the style contrast loss effectively supervising the generation of the target font.

Although FontDiffuser has achieved excellent results in generating over 100 fonts and has obtained better results for handling large deformation styles, it struggles with generating Kuzushiji fonts due to the significant differences between Kuzushiji and modern fonts, such as greater deformation and complex, consecutive stroke structures. As shown in Fig. 2, when using the Kuzushiji font image as the reference image to generate the corresponding image, FontDiffuser tends to generate the modern font rather than the Kuzushiji font. The generated result changed only for the width of each stroke and cannot make connections between the strokes. Therefore, the results did not have the style of Kuzushiji in the structure and were also accompanied by the absence of strokes. To address this

challenge, we propose KuzushijiDiffuser, a model based on FontDiffuser, specifically designed for generating Kuzushiji fonts. Below we describe our model in more detail.

Fig. 2. The results generated by FontDiffuser using Kuzushiji as the reference image.

3.2 Proposed method KuzushijiDiffuser

Based on the FontDiffuser network, we add a multiple heads stroke encoder, MLP module, and a patch content discriminator to the network. The structure of our proposed network is shown in Fig. 3. Our model is composed of three main components: the stroke style module, the content retaining module, and the conditional diffusion model. The stroke style module and content retaining module serve as generate conditions that guide the diffusion model in generating fonts. The inputs to our model are content font images and style font images. The features of the style and content images are extracted through the two modules respectively used as conditions, enabling the diffusion model to generate Kuzushiji font images.

Stroke style model FontDiffuser utilized the MCA module and the RSI module to capture features of reference and content images at different layer levels, ensuring the style features and structure features transfer to the generated font. However, the features obtained are based on the entire image, making it difficult to emphasize details such as strokes. Since Kuzushiji fonts often exhibit complex stroke structures and significant font deformation, these modules struggle to capture the desired features of Kuzushiji fonts. To address this issue and achieve Kuzushiji style transformation while preserving the structural integrity of Kuzushiji fonts, we propose a stroke-level feature style module named the stroke style model. Recently, MX-Font [21] employed multiple encoders to extract the content and style features of input images, using a component classifier to ensure that the extracted features correspond to each component of the font. This method is effective in extracting stroke-level features of a font image. Our proposed stroke extraction module is shown in Fig. 3, outlined by the blue line. Inspired by MX-Font, we introduce a multiple heads stroke encoder to extract stroke-level stylistic features and combine these stroke features with the overall font features as conditional inputs into the diffusion model network. Specifically,

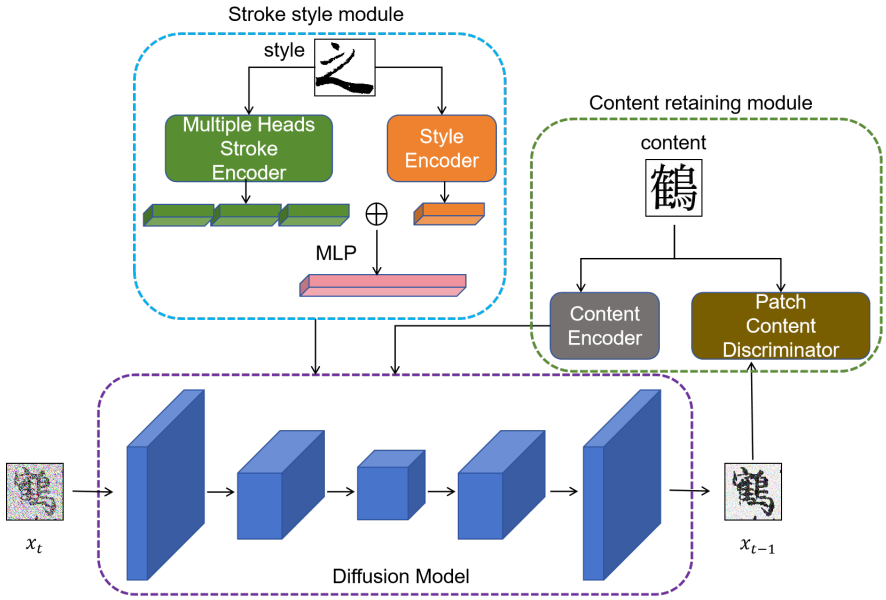

Fig. 3. Overview of our proposed method. The stroke style module and content-retaining module are represented in the blue dotted line and green dotted line boxes respectively.

the style image is passed through multiple encoders to obtain features at the stroke level. And we set the number of encoders to 4 which means we will divide every font into 4 main stroke parts. Each encoder independently encodes the style and content features at the stroke level. These style features are then concatenated to form a style feature set, while the content features are concatenated to form a content feature set. Finally, the two sets are combined to create stroke-level features. After merging the stroke-level features with the global features generated by the style encoder, we input the final features into the diffusion model as conditions through the MLP module.

Additionally, since Kuzushiji fonts are sourced from ancient books, the character images in the database are often of low resolution and accompanied by some significant noise that could influence the quality of the generated font images. To address this issue, we pre-processed the data before training the model to ensure the acquisition of as many noise-free images as possible. Specifically, we applied morphological operations to the images to eliminate noise in the fonts. The comparison of the images before and after pre-processing is shown in Fig. 4.

Content retaining model and Diffusion model In the second stage of FontDiffuser's training, the style features of the target font and various other fonts are extracted using a style extractor for contrastive learning, reinforcing the stylistic characteristics of the generated images. We added the stroke module to

Fig. 4. Original images in the Kuzushiji dataset with noise were in the first line and images obtained after our pre-processing were in the second line.

capture stroke-level features that already have sufficient stylistic features, and emphasizing stylistic features at the stroke level reduced the global information, leading to instability in the font structure. This is demonstrated in ablation studies. Moreover, the Kuzushiji font has more deformation in font structure that often exhibits significant distortions, making it challenging to align the strokes of the content font with those of the style font. To further ensure the structural integrity of the generated fonts, we introduce a font content discriminator. This discriminator computes the loss between the patch generated by our model and the content image, helping to maintain the font's structure. The content-retaining model simplifies the learning process by requiring only one single stage of training in our method.

Regarding the diffusion model, after extracting the style and content features, we input them into the diffusion model as conditional features. During the denoising process, the diffusion model then utilizes these conditional features to predict the noise at each step, ultimately generating the Kuzushiji font image.

4 Experiments

4.1 Experimental Settings and Results

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. In the inference process, we only need to arbitrarily select a Kuzushiji font image as the reference image and input the reference image and the desired content image into the model. we use the DPM-Solver++ sampler with 50 inference steps. Our inference time takes about 7 to 10 seconds.

4.2 Japanese Classical Kuzushiji Dataset

As of January 2024, the Japanese Classical Kuzushiji Dataset[2] consisting of 6,151 pages of image data from 44 classical works owned by the National Institute of

[2] http://codh.rois.ac.jp/char-shape/

Japanese Literature has been published to the public by the Center for Open Data in the Humanities (CODH). The dataset includes famous stories such as "Oragaharu" and "Amagatsu Monogatari" among other works that illustrate the culture of the Edo period. Fig.5 shows the example of images included in the Japanese Classical Kuzushiji Dataset. However, the Kuzushiji dataset has the following issues. Firstly, the data is highly imbalanced across classes. The class with the most images contains 41,293 instances, while most classes have very few images, with 790 classes represented by only a single image. Secondly, because most images were taken directly from ancient books, many of the font images are low-resolution and accompanied by significant noise and unevenly colored backgrounds, making it difficult to obtain high-quality Kuzushiji font images.

In this experiment, since we focus on the generation of Kuzushiji fonts and extracting the fonts from backgrounds would involve the text recognition task, we opted to use black-and-white text images from the dataset as our train data. Specifically, we collected about 4,300 Kuzushiji font images from "Oragaharu" to train our network.

Fig. 5. The sample page from the Japanese Classical Kuzushiji Dataset and some sample font images of Kuzushiji font. (Cited from http://codh.rois.ac.jp/)

4.3 Comparison with State-of-the-Art Method

We compare our method with the base method, FontDiffuser, and the other state-of-the-art font generation methods such as MX-Font, CF-Font, and DG-Font. Additionally, we also compare the retrained model of FontDiffuser on the Kuzushiji dataset.

The results of the qualitative evaluation are shown in Fig. 6. The original FontDiffuser could only generate modern fonts and did not allow for effective Kuzushiji font generation even using the Kuzushiji font as the reference image. Although we retrain the basic method based on the Kuzushiji dataset and get more Kuzushiji style features, the re-trained FontDiffuser model lacks the structural integrity of the font. Confusing continuous structures occur at the stroke

level, such as generating strokes that should not be continuous and strokes that should not be separated, which leads to some characters becoming unrecognizable. The results of other few-shot font generation methods like MXFont, CF-Font, and DG-Font although maintain the shape of the font well, but have less font deformation in particular stroke levels. The result of these methods generated more like the modern font and lacked the features of the Kuzushiji font. Compared to the target text, the results of our model not only align with the target text in the overall text structure but especially in the realization of the Kuzushiji style at the level of stroke structure. Even in the characters that have large deformation from the source image, our method could generate the font that has the structure with the target font. Therefore, the previous method could not generate the Kuzushiji font and our method achieved state-of-the-art performance in Kuzushji font generation.

Fig. 6. Comparison results between our method and previous state-of-the-art methods.

For quantitative evaluation, we use FID, SSIM, LPIPS, RMSE, and L1 to evaluate the results generated by each method with the target image. RMSE and L1 allow pixel-level comparisons to measure the similarity between the generated image and the target image, while LPIPS are used to discriminate the similarity of the images from the human perspective. FID evaluates the similarity of the overall distribution and SSIM could measure structural similarity between two images. Specifically, we randomly selected 100 texts for the experiment. These texts contained simple as well as complex Kanji characters. The results of the quantitative evaluation are shown in Table 1. Our method achieved the best results in FID and LPIPS and improved by over 5 percent to the second score. This shows that our results are more similar to the target font images in overall structure. We also achieved the best SSIM score which proved that our results

were closer to the target in details of the image. We also achieved the second in L1 and RMSE. Therefore, the experimental results show that our method outperforms existing methods in Kuzushiji font generation and could generate satisfactory Kuzushiji font results.

Table 1. Quantitative evaluation results with previous methods. Bold and underlined indicate the best and second-best results, respectively, and the percentage in the last line indicates the rate at which our method improves relative to the second results.

	FID↓	LPIPS↓	L1↓	RMSE↓	SSIM↑
FontDiffuser	1.4768	0.4844	**0.6071**	**0.5001**	<u>0.2988</u>
FontDiffuser(retrain)	<u>1.1811</u>	<u>0.4237</u>	0.6446	0.5217	0.2955
MX-Font	1.2392	0.4307	0.6459	0.5210	0.2400
CF-Font	1.1555	0.5274	0.7456	0.5639	0.2104
DG-Font	1.5265	0.5495	0.7226	0.5585	0.2006
Ours	**1.1219**	**0.3978**	<u>0.6417</u>	<u>0.5182</u>	**0.3029**
	5.01%	6.11%	(-)	(-)	1.37%

Fig. 7. Qualitative evaluation results of ablation studies.

4.4 Ablation Studies

We conducted ablation studies to analyze the effectiveness of each part of our approach. We conducted experiments on the model without the stroke style module and the model without the content discriminator. The results of the qualitative evaluation are shown in Fig. 7. In the baseline approach without the addition of the style module and content module, only modern text can be generated even using the Kuzushiji font images as the reference image for experimentation. After the addition of the style module, the generated result has obvious characteristics of the Kuzushiji font, but there are additions or missing in some of the strokes. After adding the content module, strokes in the text match the source text and ensure the readability of fonts. Therefore, our model could successfully generate the natural Kuzushiji font and safeguard the integrity of

the font structure. The content discriminator is intended to control the integrity of the font shape, and since the font of the baseline does not change significantly from the source font and without the Kuzushiji style when the style module is not added, we did not experiment with adding only the content discriminator.

The quantitative evaluation results are shown in Table 2. We compare the results with the target images. The results show that we achieved first scores in FID and LPIPS, and second in other scores. Therefore, the overall experimental results show that our stroke style module effectively achieves control over the style features at the stroke level, and the addition of the content discriminator ensured the structure and readability of the font.

Table 2. Ablation study results of our proposed method.

Style Module	Content Discriminator	FID↓	LPIPS↓	L1↓	RMSE↓	SSIM↑
×	×	1.3865	0.4837	**0.5934**	**0.4947**	**0.3247**
✓	×	<u>0.9160</u>	<u>0.4501</u>	0.6686	0.5261	0.3073
✓	✓	**0.9125**	**0.3907**	<u>0.6354</u>	<u>0.5249</u>	<u>0.3163</u>

5 Discussion

In modern fonts, the stroke features of the same style are consistent, meaning that the same strokes are written in the same way across different texts. However, what sets Kuzushiji different from modern text generation is that Kuzushiji often exhibits variations in writing styles even within the same text, and the same strokes in different texts may be written differently. When we incorporate stroke features into the model, the writing style of each stroke can only maintain roughly similar characteristics, but cannot be the perfect match. This makes it challenging to generate fonts that precisely match the target font. In the future, we plan to incorporate stroke order and other related features to address this issue.

6 Conclusion

In this study, we propose a diffusion model-based method that includes a stroke-level style module and a content-retaining module for generating Kuzushiji fonts. Experimental results demonstrate that our method effectively generates Kuzushiji fonts while preserving their structural integrity. Our proposed stroke-level style extractor successfully captures stroke-level stylistic features, and when combined with the content-retaining module, our method achieves state-of-the-art results in Kuzushiji font generation. This approach provides an effective method of supplementing and digitizing Kuzushiji data. However, due to the limited data available, our method currently supports only certain styles of Kuzushiji. In the future, we plan to enhance our model to support the generation of a broader range of Kuzushiji font styles.

References

1. Anoosheh, A., Agustsson, E., Timofte, R., Van Gool, L.: Combogan: Unrestrained scalability for image domain translation. In: Proc. of IEEE Computer Vision and Pattern Recognition workshops. pp. 783–790 (2018)
2. 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)
3. 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)
4. Chang, B., Zhang, Q., Pan, S., Meng, L.: Generating handwritten chinese characters using cyclegan. In: 2018 IEEE winter conference on applications of computer vision (WACV). pp. 199–207. IEEE (2018)
5. Choi, Y., Choi, M., Kim, M., Ha, J.W., Kim, S., Choo, J.: Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In: Proc. of IEEE Computer Vision and Pattern Recognition. pp. 8789–8797 (2018)
6. Choi, Y., Uh, Y., Yoo, J., Ha, J.W.: Stargan v2: Diverse image synthesis for multiple domains. In: Proc. of IEEE Computer Vision and Pattern Recognition. pp. 8188–8197 (2020)
7. 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)
8. 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)
9. Huang, X., Liu, M.Y., Belongie, S., Kautz, J.: Multimodal unsupervised image-to-image translation. In: Proc. of European Conference on Computer Vision. pp. 172–189 (2018)
10. Huang, Y., He, M., Jin, L., Wang, Y.: Rd-gan: Few/zero-shot chinese character style transfer via radical decomposition and rendering. In: Proc. of European Conference on Computer Vision. pp. 156–172. Springer (2020)
11. Hui, L., Li, X., Chen, J., He, H., Yang, J.: Unsupervised multi-domain image translation with domain-specific encoders/decoders. In: 2018 24th International Conference on Pattern Recognition (ICPR). pp. 2044–2049. IEEE (2018)
12. 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)
13. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. In: Proc. of IEEE Computer Vision and Pattern Recognition. pp. 1125–1134 (2017)
14. 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)
15. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114 (2013)
16. Liu, M.Y., Breuel, T., Kautz, J.: Unsupervised image-to-image translation networks. Advances in Neural Information Processing Systems **30** (2017)

17. Liu, M.Y., Huang, X., Mallya, A., Karras, T., Aila, T., Lehtinen, J., Kautz, J.: Few-shot unsupervised image-to-image translation. In: Proc. of IEEE International Conference on Computer Vision. pp. 10551–10560 (2019)
18. Liu, M.Y., Tuzel, O.: Coupled generative adversarial networks. Advances in Neural Information Processing Systems **29** (2016)
19. 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)
20. 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)
21. 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)
22. 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)
23. 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)
24. 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
25. Sun, D., Ren, T., Li, C., Su, H., Zhu, J.: Learning to write stylized chinese characters by reading a handful of examples. arXiv preprint arXiv:1712.06424 (2017)
26. 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)
27. 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)
28. 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)
29. 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)
30. 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)
31. Zhang, Y., Zhang, Y., Cai, W.: Separating style and content for generalized style transfer. In: Proc. of IEEE Computer Vision and Pattern Recognition. pp. 8447–8455 (2018)