Few shot font generation via transferring similarity guided global style and quantization local style

Wei Pan [1] Anna Zhu* [1] Xinyu Zhou [1] Brian Kenji Iwana [2] Shilin Li [1]

[1]School of Computer Science and Artificial Intelligence, Wuhan University of Technology [2]Human Interface Laboratory, Kyushu University

ICCV23 PARIS

Problem definition and motivation

Font design techniques can benefit many critical applications, such as logo designs, data augmentation for text-related tasks etc. We uses deep learning technology as shown in Fig 1 to reduce manual design costs.

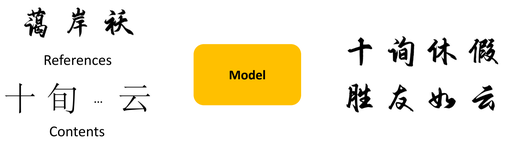

Figure 1. The model receives some reference characters (3 etc. providing the style) and target content characters (usually using Song or Kaiti) to generate a full font library.

Our contribution can be summarized as:

- We propose a novel FFG method leveraging complementary global and local representations. In this way, the quality of model generation results as shown in Fig 2 is improved.
- The local feature representation comes from the slef-learned discrete vector codebook form the pre-trained VQ-VAE. Global style feature is adaptively re-weighted the reference style feature by calculating the glyph similarity of content-reference characters.
- The experimental results of the model on the Chinese data set have reached SOTA. Our method can be trained on other languages without any modification, and can also be zero-shot.

Representation of style features	Generate results					
Global Representation	御	僻	拌	乳	咙	游
Local Representation	御	僻	拌	乳	咙	游
Component label	彳+卸	亻尸口立十	扌+半	爫子乚	口+龙	氵方𠂉子
Global + Local Fusion(ours)	御	僻	拌	乳	咙	游
Ground truth	御	僻	拌	乳	咙	游

Figure 2. Compared with methods that only use global or local style feature representation, our method improves the generation of local details without requiring additional label information.

Pre-trained VQ-VAE

The content encoder E_c and component Codebook are obtained through pre-training via VQ-VAE framework.

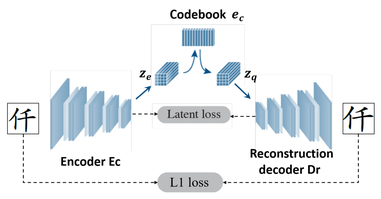

Figure 3. The glyph feature decomposing network for pre-training the content encoder and obtaining component representation.

Few shot font generation model architecture

The FFG model in this method is shown as Fig 4. It consists of a generator G and a multi-task discriminator D. The generator G includes the content encoder E_c (pre-trained) and the style encoder E_r. The CAM module is used to obtain the stylized component feature representation, the GSA module is used to obtain the global style feature representation, and the Decoder is used to fuse features and obtain the target image.

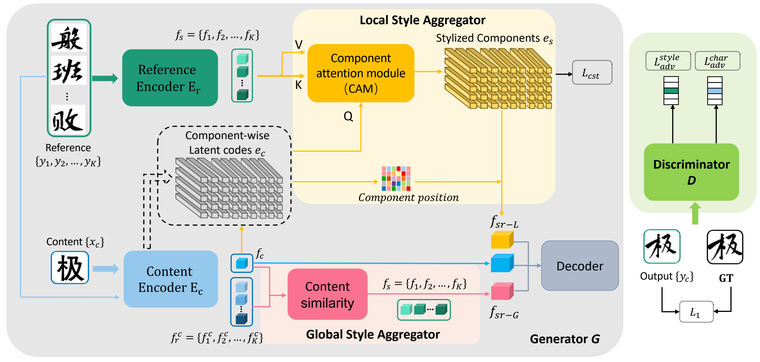

Figure 4. The architecture of our model. The generator consists of five parts: a pre-trained content encoder, a reference style encoder (marked in dark green), a local style aggregator via CAM (marked in yellow), a global style aggregator with content similarity guidance (marked in pink), and a decoder combining content and style features and style features for font generation (marked in gray). A discriminator (marking green) is followed to distinguish the real and fake images, and it simultaneously classifies the content and style category of the generated character.

Loss functions

We optimize the FFG model by the following full objective function:

$$\min_{E_s,G}\max_{D}\mathcal{L}_{adv}^{D}+\mathcal{L}_{adv}^{G}+\lambda_1\mathcal{L}_{img}+\lambda_2\mathcal{L}_{feat}+\lambda_3\mathcal{L}_{cst}. \quad (1)$$

L_{adv} means hinge GAN loss, L_{img} and L_{feat} is the Matching loss to make the model learn pixel-level and feature-level consistency with ground truth, L_{cst} is a style contrastive loss to learn the local styles in an unsupervised way. The λ_1, λ_2 and λ_3 is weighting hyperparameter.

Experiment result

FFG experimental results on Chinese data set

We tested our method on the Chinese dataset with kshot=3 and conducted comparative experiments with other baseline methods.

Figure 5. FFG results of each method on UFUC and UFSC dataset. We represent the generated samples of five different kinds of fonts, given three references and three content images per font. The red boxes represent the better generation details.

More experiment result

Train and test our FFG method on the Japanese dataset.

Our method is based on a self-learning strategy, does not require any pre-annotation, and can be quickly used for training and inference in other languages.

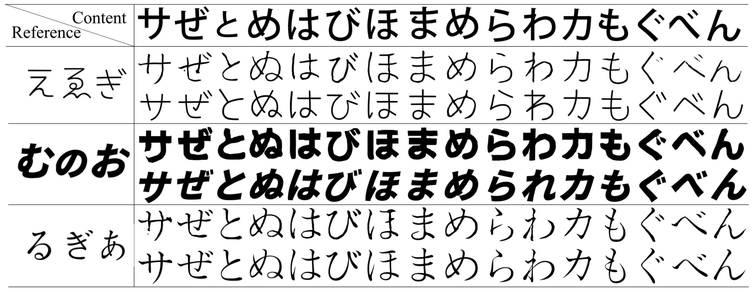

Figure 6. FFG results of Japanese scripts. For each style, the upper and lower rows respectively represent the model generation results and GT.

zero-shot inference results

In addition, considering that the strokes and spatial layout of characters such as Korean and Japanese are less complex than Chinese characters, the parameters trained on the Chinese data set can be directly used for inference in other languages.

Figure 7. Cross-language generation and Chinese generation comparison

Other informations

Paper

Code

{aaawei, annazhu}@whut.edu.cn