MetaScript: Few-Shot Handwritten Chinese Content Generation via Generative Adversarial Networks

Project of AI3604 Computer Vision, 2023 Fall, SJTU

Jiazi Bu, Qirui Li, Kailing Wang, Xiangyuan Xue, Zhiyuan Zhang

February 6, 2024

SEIEE, SJTU

Introduction

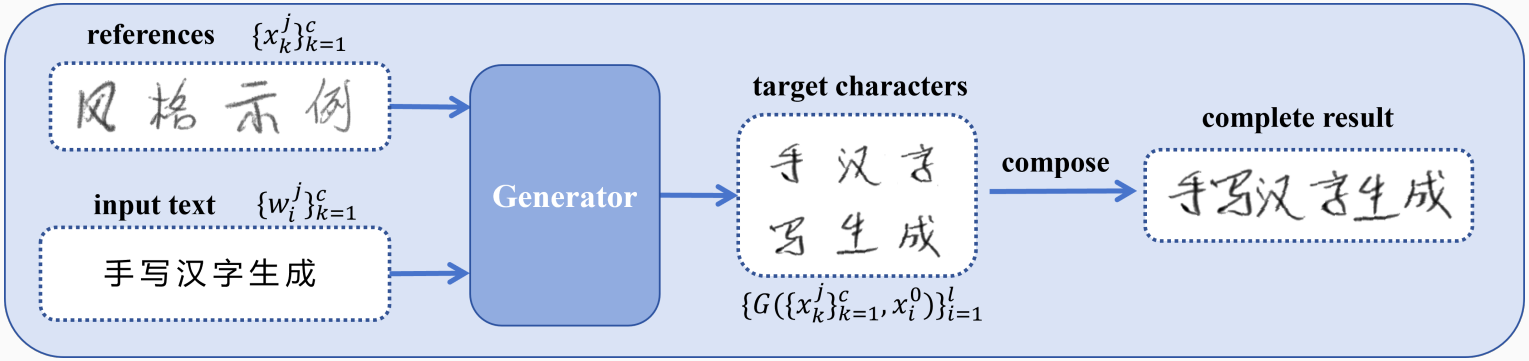
Introduction

Motivation

- Although we have entered the era of information technology, handwriting is still an important way of communication. In some formal occasions, handwritten contents are required.
- From time to time we are tired of writing large amounts of Chinese contents by hand. An automatic system is urgently needed to relieve our hands.

Introduction

Problem Definition



- Given some style references and a content text, the system should generate the corresponding handwritten Chinese characters with the given style. The characters should be naturally arranged like a human-written content.

Method

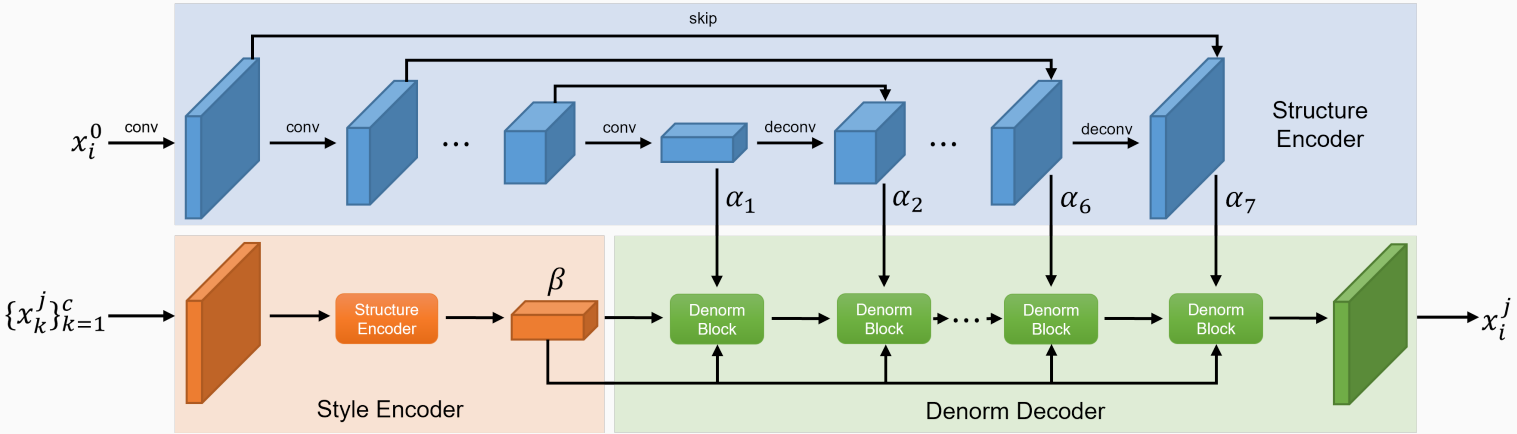
Method

Overall Pipeline

- Let $x_i^j \in \mathcal{D}$ denote the character with the i-th type written by the j-th writer where $1 \leq i \leq n$ and $1 \leq j \leq m$.
- Train a character generator G. Given some references $\{x_k^j\}_{k=1}^c$ and a template x_i^0, the generated result $G(\{x_k^j\}_{k=1}^c, x_i^0)$ should be similar to x_i^j, which inherits the structure of the i-th type and the style of the j-th writer.
- Build a Chinese content generating system S by cascading G with a character composer C, which stitches the generated characters into a handwritten style content.

Method

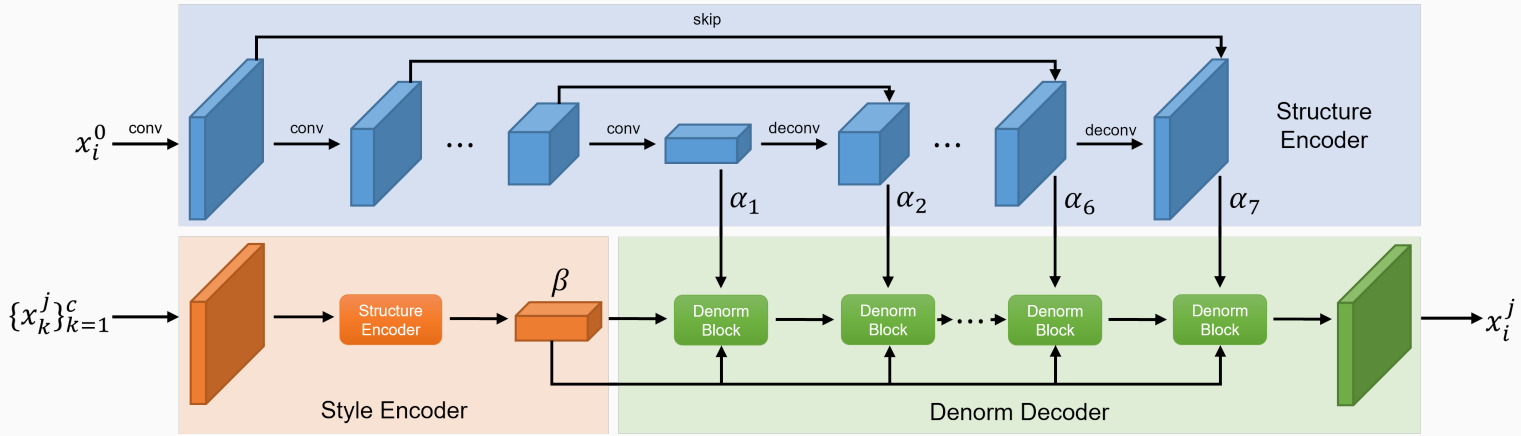
Character Generator: Structure Encoder



- The structure encoder E_α follows the U-Net architecture, which includes 6 down-sampling blocks and 6 up-sampling blocks, extracting 7 feature maps $\alpha_1, \alpha_2, \ldots, \alpha_7$ with different scales from the template x_i^0.

Method

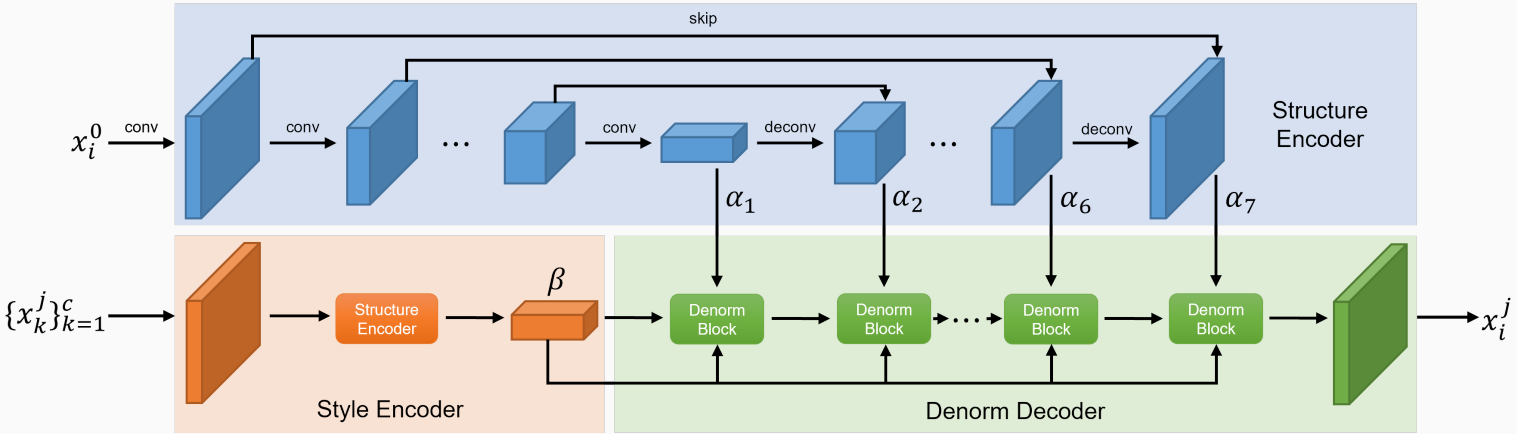
Character Generator: Style Encoder



- The style encoder E_β applies the ResNet-18 architecture with input channels modified to c and a linear layer added at the end, which extracts a 512-dimensional feature vector β from the references $\{x_k^j\}_{k=1}^c$.

Method

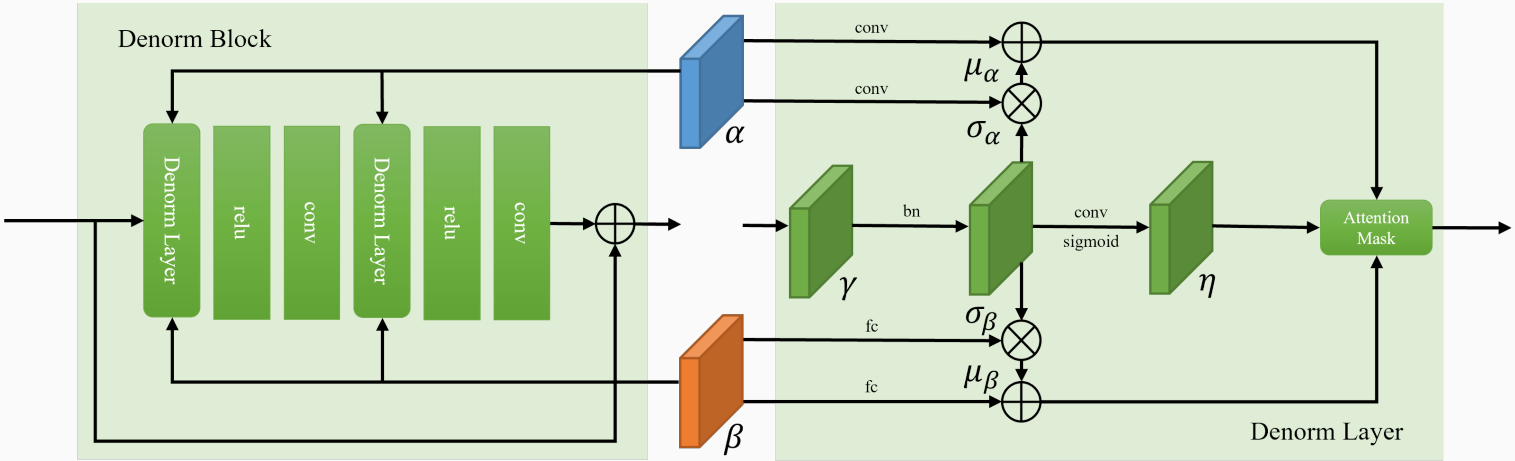
Character Generator: Denormalization Decoder



- Both the structure information $\alpha_1, \alpha_2, \ldots, \alpha_7$ and the style information β will be fed into the denormalization decoder D_γ, which is composed of 7 cascaded denormalization blocks $D_\gamma^1, D_\gamma^2, \ldots, D_\gamma^7$.

Method

Character Generator: Denormalization Block



- The denormalization layer is composed of a normalization step, a denormalization step and an attention mechanism. The denormalization block is composed of a skip connection and two identical layers.

Method

Character Generator: Denormalization Layer

- Normalization

$$\bar{\gamma} = \frac{\gamma - \mu_\gamma}{\sigma_\gamma}$$

Denormalization

$$\hat{\alpha} = \sigma_\alpha \times \bar{\gamma} + \mu_\alpha$$
$$\hat{\beta} = \sigma_\beta \times \bar{\gamma} + \mu_\beta$$

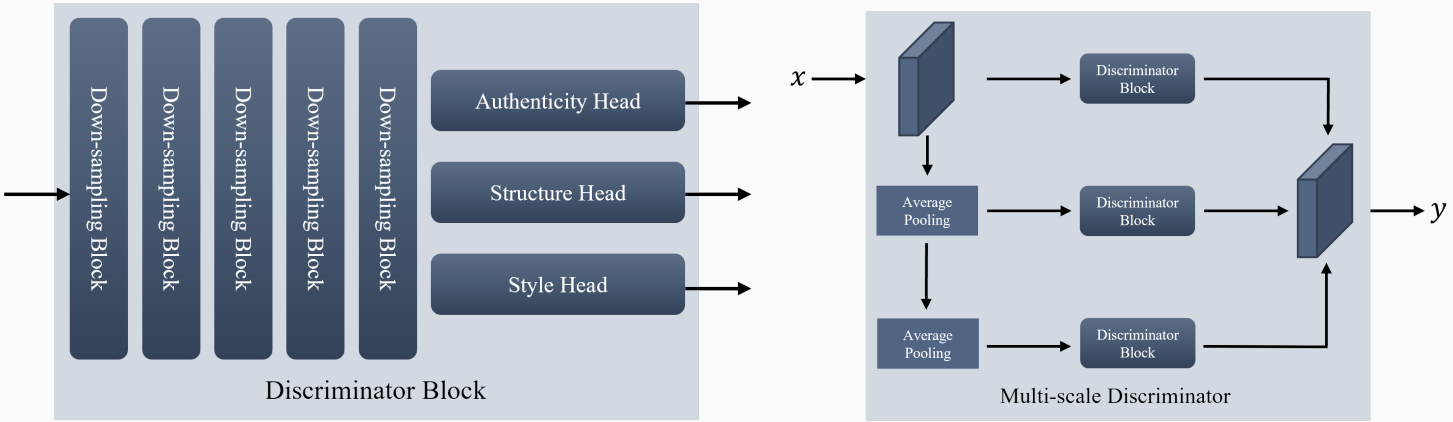
Attention

$$\hat{\gamma} = (1 - \eta) \times \hat{\alpha} + \eta \times \hat{\beta}$$

- The key idea of the denormalization layer is to adaptively adjust the effective regions of the structure feature map and the style feature map, so that the generated character can inherit the structure of the template and the style of the references.

Method

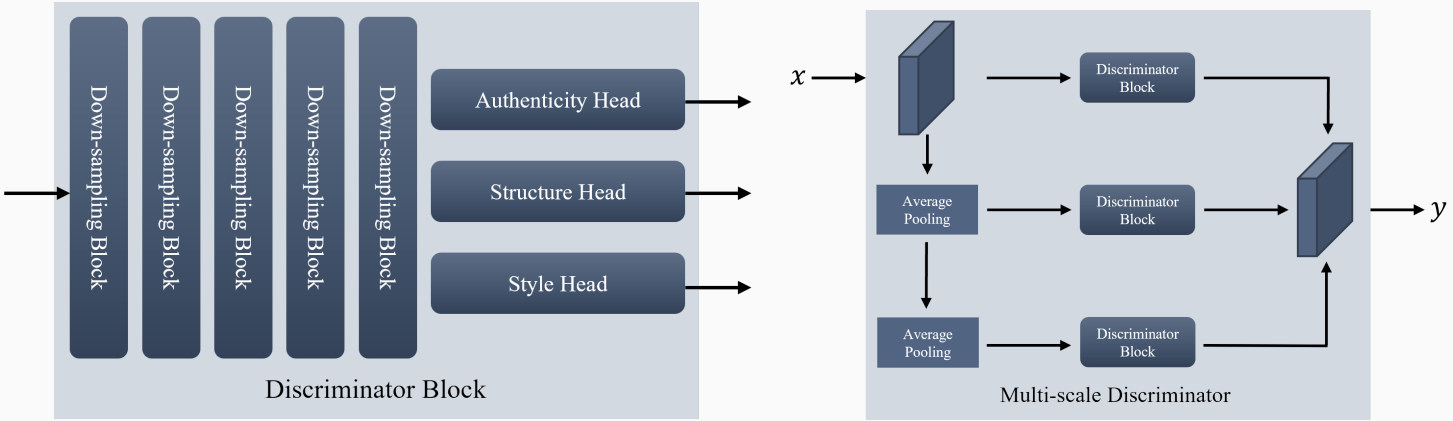
Multi-scale Discriminator: Discriminator Block



- The discriminator block is composed of 5 down-sampling blocks and 3 classification heads to to predict authenticity, type and writer all together.

Method

Multi-scale Discriminator: Discriminator Framework



- The discriminator D is composed of 2 average pooling layers and 3 discriminator blocks. The input character x will be down-sampled to form 3 different scales and fed into the corresponding discriminator blocks.

Method

Training Objective: Overall Loss

- The overall loss function is composed of 5 parts: adversarial loss $\mathcal{L}_{adv}$, classification loss $\mathcal{L}_{cls}$, structure loss $\mathcal{L}_{str}$, style loss $\mathcal{L}_{sty}$ and reconstruction loss $\mathcal{L}_{rec}$.

$$\mathcal{L}_{all}^{G} = \lambda_{adv}^{G}\mathcal{L}_{adv}^{G} + \lambda_{cls}^{G}\mathcal{L}_{cls}^{G} + \lambda_{str}^{G}\mathcal{L}_{str}^{G} + \lambda_{sty}^{G}\mathcal{L}_{sty}^{G} + \lambda_{rec}^{G}\mathcal{L}_{rec}^{G}$$
$$\mathcal{L}_{all}^{D} = \lambda_{adv}^{D}\mathcal{L}_{adv}^{D} + \lambda_{cls}^{D}\mathcal{L}_{cls}^{D}$$

- Note that λ_{adv}^{G}, λ_{cls}^{G}, λ_{str}^{G}, λ_{sty}^{G}, λ_{rec}^{G}, λ_{adv}^{D} and λ_{cls}^{D} are the hyperparameters to balance the loss functions.

Method

Training Objective: Adversarial Loss

- The adversarial loss is to train the discriminator D to distinguish the generated character $G(\{x_k^j\}_{k=1}^c, x_i^0)$ from the real character x_i^j, which indirectly encourages the generator G to generate more plausible characters.

$$\mathcal{L}_{adv}^G = -\sum_{s=1}^{3} \log D_\phi^s(G(\{x_k^j\}_{k=1}^c, x_i^0))$$

$$\mathcal{L}_{adv}^D = -\sum_{s=1}^{3} \log D_\phi^s(x_i^j) - \sum_{s=1}^{3} \log[1 - D_\phi^s(G(\{x_k^j\}_{k=1}^c, x_i^0))]$$

Method

Training Objective: Classification Loss

- The classification loss is to train the discriminator D to precisely predict the type and writer of the given character x, which indirectly encourages the generator G to generate characters with accurate structure and style.

$$\mathcal{L}_{cls}^{G} = -\sum_{s=1}^{3} \log D_{\alpha}^{s}(G(\{x_k^j\}_{k=1}^{c}, x_i^0)) - \sum_{s=1}^{3} \log D_{\beta}^{s}(G(\{x_k^j\}_{k=1}^{c}, x_i^0))$$

$$\begin{aligned}\mathcal{L}_{cls}^{D} = &-\sum_{s=1}^{3} \log D_{\alpha}^{s}(x_i^j) - \sum_{s=1}^{3} \log D_{\beta}^{s}(x_i^j) \\ &-\sum_{s=1}^{3} \log[D_{\alpha}^{s}(G(\{x_k^j\}_{k=1}^{c}, x_i^0))] - \sum_{s=1}^{3} \log[D_{\beta}^{s}(G(\{x_k^j\}_{k=1}^{c}, x_i^0))]\end{aligned}$$

Method

Training Objective: Structure Loss

- Intuitively, we expect that the generated character $G(\{x_k^j\}_{k=1}^c, x_i^0)$ can inherit the structure of the template x_i^0. Therefore, The feature maps $\alpha_1, \alpha_2, \ldots, \alpha_7$ extracted by the structure encoder E_α should be invariant.
- The structure loss not only encourages the generator G to generate the correct structure, but also encourages the structure encoder E_α to extract valid structure features.

$$\mathcal{L}_{str}^G = \frac{1}{2}\|E_\alpha(G(\{x_k^j\}_{k=1}^c, x_i^0)) - E_\alpha(x_i^0)\|_2^2$$

Method

Training Objective: Style Loss

- We also expect that the generated character $G(\{x_k^j\}_{k=1}^c, x_i^0)$ can follow the style of the references $\{x_k^j\}_{k=1}^c$. Therefore, the feature vector β extracted by the style encoder E_β should be invariant.
- The style loss not only encourages the generator G to generate the correct style, but also encourages the style encoder E_β to extract valid style features.

$$\mathcal{L}_{sty}^G = \frac{1}{2}\|E_\beta(G(\{x_k^j\}_{k=1}^c, x_i^0)) - E_\beta(\{x_k^j\}_{k=1}^c)\|_2^2$$

Method

Training Objective: Reconstruction Loss

- The reconstruction loss measures the pixel-wise difference between the generated character $G(\{x_k^j\}_{k=1}^c, x_i^0)$ and the ground truth x_i^j, which is a direct supervision.

$$\mathcal{L}_{rec}^G = \| G(\{x_k^j\}_{k=1}^c, x_i^0) - x_i^j \|_1$$

- Note that the reconstruction loss is not to force the generated character to be exactly the same with the ground truth because such constraint will limit the diversity of the generated characters.

Method

Content Composition

- We develop a typesetting tool, Typewriter, for reorganizing the output characters from the generator into a complete result.
- A random transformation is applied to mimick the alignment effect of human handwriting.

Algorithm 1: Typewriter Procedure

Input : style references $\{x_k\}_{k=1}^{c}$, content text $\{w_i\}_{i=1}^{l}$, expected character size $size_c$, expected line width $width_l$

Output : arranged handwritten content image I_a

Initialize: $cursor \leftarrow$ position 0, $I_a \leftarrow$ empty image;
for w_i *in* $\{w_i\}_{i=1}^{l}$ **do**
  if w_i *is line break* **then**
    move $cursor$ to next line;
    continue;
  else if w_i *is space* **then**
    $g_i \leftarrow$ empty space with $\frac{size_c}{2}$ width;
  else if w_i *is character* **then**
    generate $g_i \leftarrow G(\{x_k\}_{k=1}^{c}, x_{w_i}^{0})$ with $size_c$;
  else if w_i *is punctuation* **then**
    $g_i \leftarrow$ punctuation template;
  end
  apply random transformation on g_i;
  plot g_i at $cursor$ in I_a and update $cursor$;
  if $cursor > width_l$ **then**
    move $cursor$ to next line;
  end
end
return handwritten content image I_a

Experiments

Experiments

Experiment Setup: Dataset Construction

- The training set is build based on the CASIA-HWDB-1.1 dataset. We select approximately 1M script images from 300 writers, including 3755 common Chinese characters
- By preprocessing, the script images are resized into 128×128 resolution, named by the types and classified by the writers, which will be used as style references.
- We also render the prototype images of the characters with the same resolution from the Source Han Sans CN font, which will be used as structure templates.

Experiments

Experiment Setup: Implementation Details

- The model is trained for 100k iterations with a batch size of 32 on a single NVIDIA GeForce RTX 3090 GPU.
- We use the Adam optimizer for training. The learning rate is set to 0.0001 for both the generator and the discriminator.
- The weights of the loss function are $\lambda_{\text{adv}}^{G} = 1$, $\lambda_{\text{cls}}^{G} = 1$, $\lambda_{\text{str}}^{G} = 0.5$, $\lambda_{\text{sty}}^{G} = 0.1$, $\lambda_{\text{rec}}^{G} = 20$, $\lambda_{\text{adv}}^{D} = 1$ and $\lambda_{\text{cls}}^{D} = 1$.

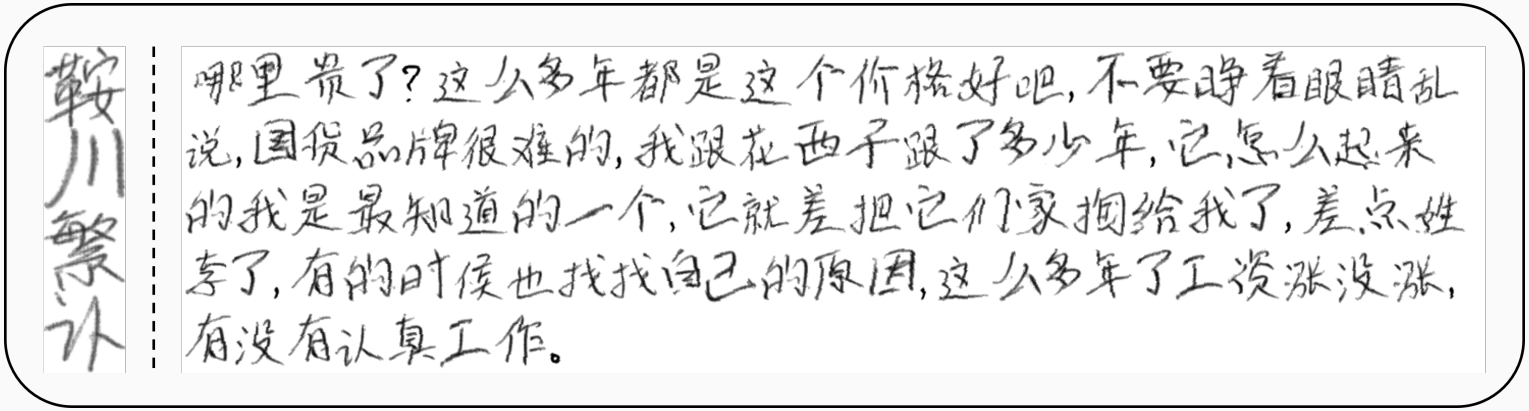
Experiments

Character Results

- The first 4 rows are the style references. The 5th row is the structure template. The 6th row is the ground truth. The last row is the generated character.
- The generated characters show clear structures and explicit styles, which are similar to the ground truth.

Experiments

Content Results



- The left hand side shows the style references. The right hand side shows the generated results.
- The generated content follows the style references and is synthesized in a natural way, which looks like a handwritten content by a human.

Experiments

Ablation Studies: Evaluation Metrics

- We apply recognition accuracy (RA), inception score (IS) and Frechet inception distance (FID) to evaluate the quality and diversity of the generated results.
- We use an EfficientNetV2 model pretrained on the CASIA-HWDB-1.1 dataset to extract high-level features and perform low-level predictions for computation.
- The test set is selected from the CASIA-HWDB-1.0 dataset, which contains 1000 characters written by 10 writers.

Experiments

Ablation Studies: Reference Number

- To figure out how the number of references affects the performance of the model, we train the model with 1, 2, 4 and 8 references for 100k iterations respectively.

Reference	RA↑	IS↑	FID↓
1	79.8%	55.543	197.950
2	77.5%	58.524	190.193
4	**81.9%**	58.172	**187.365**
8	76.2%	**58.598**	188.411

Experiments

Ablation Studies: Reference Number

- In general, as the number of references increases, the inception score increases and the Frechet inception distance decreases, which indicates better authenticity and diversity of the generated results.
- This is reasonable because more references provide more sufficient style information for the generator.

Reference	RA↑	IS↑	FID↓
1	79.8%	55.543	197.950
2	77.5%	58.524	190.193
4	**81.9%**	58.172	**187.365**
8	76.2%	**58.598**	188.411

Experiments

Ablation Studies: Loss Function

- To figure out whether each loss term is necessary, we train the model with 4 references for 100k iterations, but each loss term is removed respectively.

Loss	RA↑	IS↑	FID↓
w/ $\mathcal{L}_{all}$	81.9%	58.172	187.365
w/o $\mathcal{L}_{adv}$	0.0%	1.098	287.104
w/o $\mathcal{L}_{cls}$	0.1%	1.601	318.286
w/o $\mathcal{L}_{str}$	84.4%	57.578	181.738
w/o $\mathcal{L}_{sty}$	67.4%	55.108	204.016
w/o $\mathcal{L}_{rec}$	72.3%	54.316	202.782

Experiments

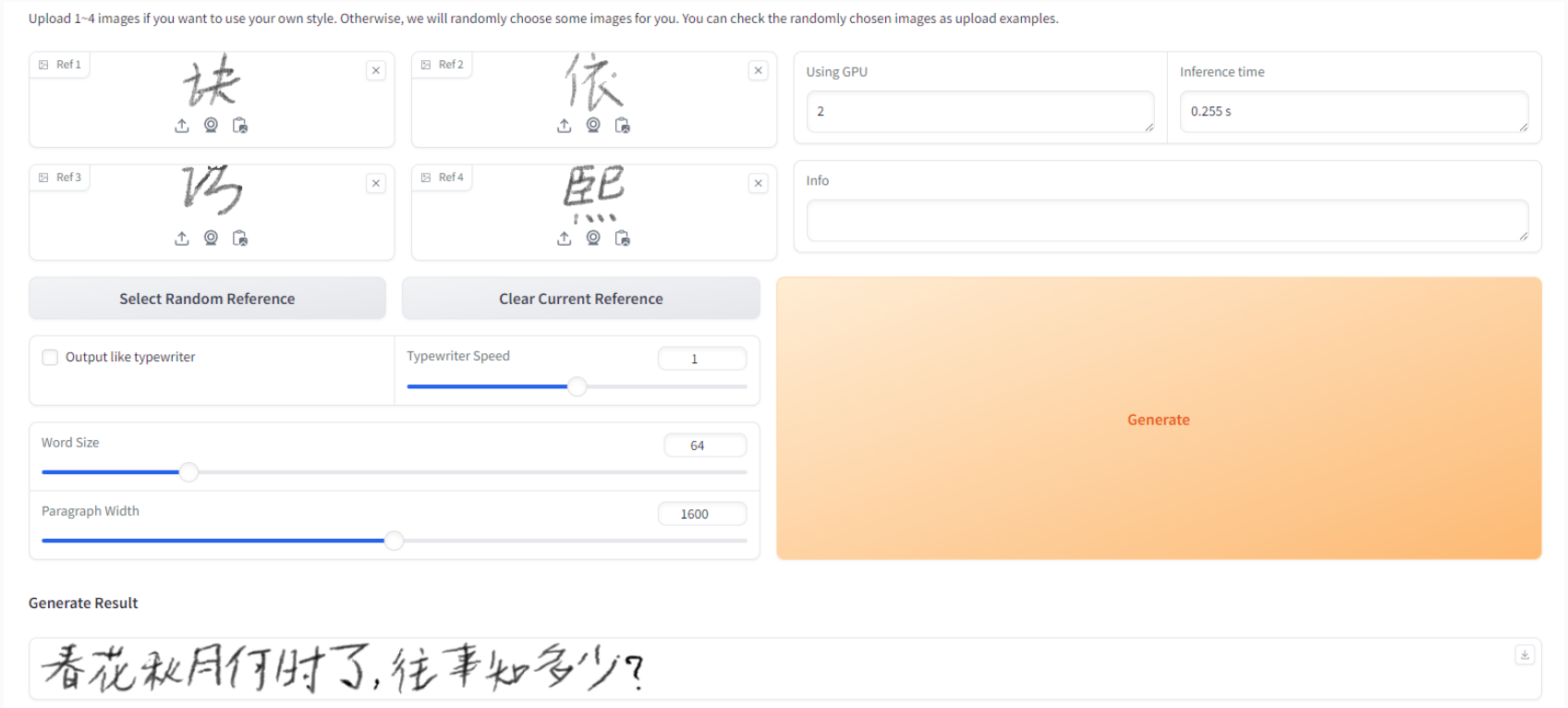
Ablation Studies: Loss Function

- The models with one of the loss terms removed show performance degradation to varying degrees.
- Removing $\mathcal{L}_{sty}$ and $\mathcal{L}_{rec}$ leads to a modest decrease in performance, while removing $\mathcal{L}_{adv}$ and $\mathcal{L}_{cls}$ results in a complete failure.

Project Demo

▸ Demo Website

website: https://www.loping151.com link: cv2023 demo



Thank You!