DSE-NET: ARTISTIC FONT IMAGE SYNTHESIS VIA DISENTANGLED STYLE ENCODING

Xiang Li, Lei Wu, Xu Chen, Lei Meng*, Xiangxu Meng*

School of Software, Shandong University, China
202035260@mail.sdu.edu.cn, i_lily@sdu.edu.cn,
listening@mail.sdu.edu.cn, lmeng@sdu.edu.cn, mxx@sdu.edu.cn

ABSTRACT

Recently, the artistic font generation has made significant progress. However, existing methods typically treat the style of artistic font as a whole. Their performance is usually limited to the artistic fonts with complex style elements in glyph and text effect. To solve these problems, this paper presents a disentangled style encoding network, termed DSE-Net, to synthesize artistic fonts. In order to obtain the disentangled text effect features, we introduce a perspective transformation network. We propose a cross-layer fusion mechanism to improve the artistic fonts' structure and texture according to their different representations in CNN. Notably, encoding different style elements for artistic font generation is a new task, so there is no publicly-accessible dataset. Therefore, a new dataset, termed SSAF, has been constructed. Extensive experiments demonstrate that our model significantly outperforms the state-of-the-art methods, with more fine-grained text effect and accurate stroke details.

Index Terms— Neural Style Transfer, Text Effect Transfer, Artistic Font Generation

1. INTRODUCTION

The application scene of artistic fonts can be viewed everywhere, such as advertisements, web pages, and posters. Some successful artistic fonts often attract people's attention through their elegant glyphs and exquisite text effects. However, batch production of artistic fonts is a tedious and repetitive work, which needs to apply text effects to all font images manually. Existing works have made valuable efforts in the field of artistic font generation. These efforts can be roughly divided into two aspects, some works implement text effects transfer with glyph variation, and the others only transfer the text effect without glyph variation. For the glyph-invariance task, T-Effect [1] and TET-GAN [2] introduce a distribution-aware strategy to achieve the text effects transfer. However, their works are not appropriate for the scenes of switching glyphs. For the glyph-variance task, the recent unified solution treats the glyph style and text effect style as a whole, and adopts the integrated style encoding method to learn the mixed style [3, 4, 5]. However, these methods couple the glyph style and the text effect style, which often lead to unclear textures and rough lines, as shown in Fig.1.

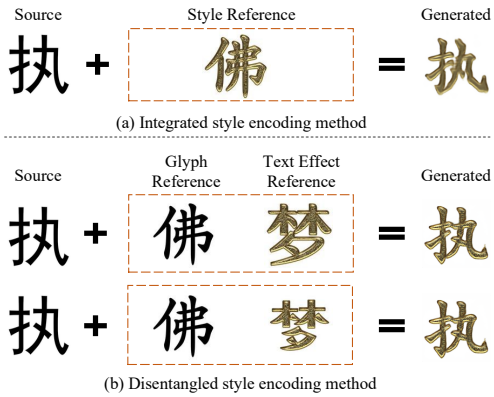


Fig. 1. (a) The existing method of using integrated style couples the glyph style and text effect style. (b) Our method uses the disentangled glyph style elements and text effect style elements to synthesize artistic fonts, with more delicate textures and more precise strokes.

We have collected a large number of artistic font images and summarized their characteristics. Their glyph is shape-based information related to the stroke tips, joined-up writing, and thickness of strokes. Their text effect is expressed as the effects based on glyphs, including color gradients, highlights, shadows, reflections, glows, pattern overlay, and other creative effects. Therefore, glyph and text effect are two independent attributes of artistic fonts, but the integrated style encoding method treats them as a whole. We suspect that this is the reason for the poor generation.

Therefore, we attempt to utilize disentangled glyph elements and text effect elements to synthesize artistic font images. Then, we propose a disentangled style encoding network (DSE-Net), which has three main modules: the perspective transformation network, the glyph encoder, and the generator. The perspective transformation network will extract the text effect from the artistic font images. To avoid the intermittent phenomenon of font strokes caused by inconsis-

*Corresponding author

tent glyph information between the text effect reference and the target artistic font, we propose the text effects consistency loss to ensure the extraction of pure text effects. The glyph encoder learns the glyph style representations from the glyph reference. Moreover, the generator learns the content feature representation of the source images, that is, the basic writing trajectory of the characters. So, the generated artistic font image can retain the content feature of the source image. In addition, for synthesizing fine-grained artistic fonts, we propose a novel cross-layer fusion mechanism to infuse such style information into the source image representation in different phases of the style transfer process. After detailed statistics, we found that the existing artistic font datasets cannot meet the disentangled style encoding task. Therefore, we construct a new style separable artistic fonts (SSAF) dataset. To the best of our knowledge, it is the first artistic font dataset that supports style separability. It can provide independent glyph reference images and text effect reference images. Moreover, it can also serve as a benchmark for Chinese and English artistic font generation tasks.

In summary, our contributions are threefold:

- We explore a new angle, the disentangled style encoding for artistic fonts, and propose a new model, DSE-Net, which solves the problem that the fine-grained style of artistic fonts is difficult to synthesize.
- We construct a new dataset, SSAF, with a large number of Chinese and English artistic fonts. It is the first artistic fonts dataset that supports style separability.
- Extensive experiments have validated the effectiveness of the disentangled style encoding and demonstrated that the DSE-Net obtains state-of-the-art performance, especially on the complex glyphs and text effects.

2. RELATED WORKS

2.1. Neural Style Transfer

Neural Style transfer is the task of migrating styles from a style image to a content image, which is closely related to texture synthesis. In the seminal work [6, 7, 8], the authors propose that the style of an image is information independent of position. They use a pre-trained VGG network to extract style features and content features. However, it relies on an optimization process, which is prohibitively slow. Recently, [9, 10] achieve high-quality style transfer by capturing the detailed features of images.

2.2. Font Generation

Font generation usually refers to glyph style transfer for obtaining a large-scale font library [11, 12, 13, 14, 15, 16]. Recently, [12, 13] separated the representations of character content and font style by introducing a generalized style transfer network, which supported generalizing generated fonts to new styles.

2.3. Artistic Font Synthesis

The synthesis of artistic characters means that given some reference styles, including glyph and text effect, some exquisite variations are automatically generated on some unseen characters. [2] first proposed a deep neural network to realize the text effects transfer. [3, 17] realizes the transfer of glyph and text effect style of English artistic fonts. There is also some work [5, 18] that proposes the integrated style transfer of English and Chinese artistic fonts. Then, [4] use an adaptive instance normalization to introduce the target integrated style as affine parameters in their artistic font style transfer task.

3. DISENTANGLED STYLE ENCODING NETWORK

The overview of the proposed model DSE-Net is shown in Fig.2. It contains three essential sub-networks: (1) The perspective transformation network. It solves the glyph information disturbance problem from the text effect reference samples. (2) The glyph encoder E_y. It learns the glyph features (e.g., geometric contours, lines et al.) and extracts the glyph style vector to guide the synthesis of the target artistic fonts. (3) The artistic font generator G. A source image is fed to G, and G extracts its content features and outputs an artistic font image conditioned on the combined glyph and text effect vectors. In addition, we propose a cross-layer fusion mechanism to fuse such style information into the source image representation in different up-sampling stages of the G.

3.1. Perspective Transformation Network

The perspective transformation network comprises a text effect encoder E_x and a perspective transformation module(PTM). When the E_x is extracting the text effect features from the text effects reference samples, the glyphs of these reference samples can be arbitrary. However, inconsistent glyph information between the text effect reference and the target artistic font will inevitably affect the correct glyph details (see Section 4.4.3). To solve this problem, we introduce the perspective transformation module and the text effects consistency loss.

Specifically, before the text effect reference samples are fed into the E_x, we add a PTM for these samples. The random perspective rotating will change the direction and width of the strokes, but the distribution and consistency of the text effect have been maintained, see Fig.2. Then, the augmented samples X^* and the original samples X have consistency in color and texture, but their position, line, and direction information have been changed. The text effect consistency loss will optimize E_x by minimizing latent geometrical feature differences

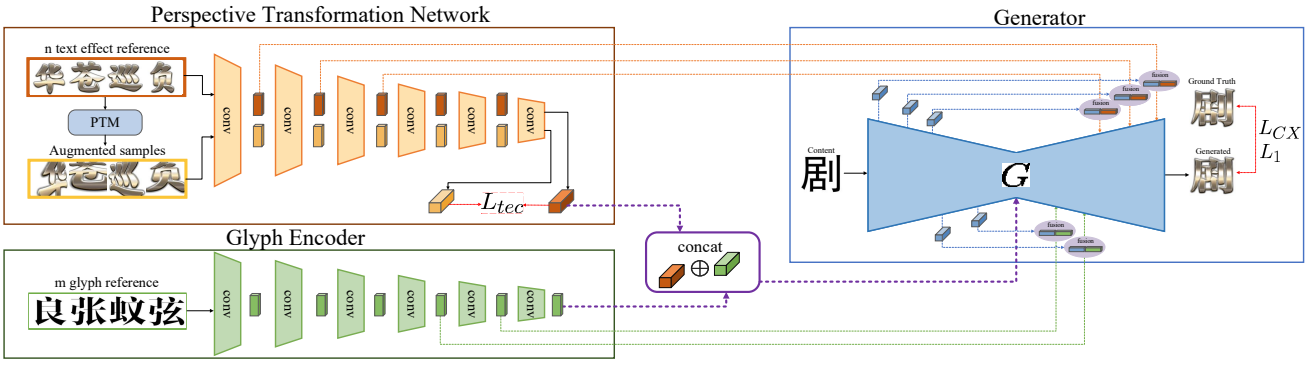


Fig. 2. Overview of the DSE-Net. In the perspective transformation network, there is a text effect encoder and a perspective transformation module. The orange, green, and blue dashed lines represent the three data stream channels established by the cross-layer fusion mechanism, which connects the text effect encoder, glyph encoder, and generator.

from X^* and X to achieve the extraction of pure text effects:

$$\theta_x^* = \underset{\theta_x}{arg\,min} \left\| E_x(X^*) - E_x(X) \right\|_1, \tag{1}$$

where $E_x(X^*)$ and $E_x(X)$ are estimated features from X^* and X. θ_x^* and θ_x are both the parameters of the E_x.

3.2. Cross-Layer Fusion Mechanism

The glyph style of the artistic fonts is a kind of overall structural feature, and the text effect reflects the distribution of artistic effects. Gatys et al. [7] point out that, at the high level of the network, detailed pixel information is lost, but some abstract features are more easily retained. Based on these observations and the previous conclusion, we propose the cross-layer fusion mechanism. It treats glyphs as the high-level features, and text effects as the low-level features. Between the up-sampling and the down-sampling process, it constructs three channels and inputs content, glyph, and text effect information into the up-sampling process. This allows the model to learn effective and sufficient information. Section 4.4.1 demonstrates the validity of this structure and view.

Specifically, in the G, the content features $f_{c,l}$ generated by the l-th layer of the down-sampling are fed into the $(d-l+1)$-th layer of the up-sampling, where d denotes the total number of layers of the up-sampling. Meanwhile, the $(d-l+1)$-th layer's input of the up-sampling is a concatenation of three feature vectors, which contains the $f_{c,l}$, the output features o_{d-l} from the previous layer, and the text effect features $f_{x,l}$ or the glyph features $f_{y,l}$. The output of the $(d-l+1)$-th up-sampling layer is formulated as:

$$o_{d-l+1} = \begin{cases} F_{d-l+1}(o_{d-l}, f_{c,l}, f_{x,l}) & , 0 < l \leq h \\ F_{d-l+1}(o_{d-l}, f_{c,l}, f_{y,l}) & , h < l \leq d \end{cases} \tag{2}$$

where h denotes a threshold that we use to measure the depth of the network, and F_{d-l+1} is the function of the $(d-l+1)$-th up-sampling layer.

3.3. Objective Function

Full Objective The objective function of our model consists of four terms: the pixel loss, the contextual loss, the text effect consistency loss, and the adversarial loss:

$$L(G) = \lambda_1 L_1 + \lambda_{CX} L_{CX} + \lambda_{tec} L_{tec} + \lambda_{adv} L_{adv}, \tag{3}$$

where $\lambda_1, \lambda_{CX}, \lambda_{tec}, \lambda_{adv}$ are hyperparameters.

L_1 Loss The L_1 loss constrains the generated image to be close to the ground truth from the pixel level:

$$L_1 = \mathbb{E} \left\| z - \hat{z} \right\|_1, \tag{4}$$

where $\hat{z}$ denotes the generated image and z denotes the ground truth.

Contextual Loss The L_1 loss assumes that the generated images and ground truth are spatially aligned and compares pixels at corresponding locations. If the synthesized image is not precisely spatially aligned to the ground truth (e.g., a small displacement or rotation), the pixel loss will be high, but the generated result is often visually acceptable. Therefore, we apply the contextual loss [19], which treats images as a collection of features, and measures the similarity between images features, ignoring the spatial location of features:

$$L_{CX}(z, \hat{z}, l) = -log(CX(\Phi^l(z), \Phi^l(\hat{z}))), \tag{5}$$

where the $\Phi^l(\cdot)$ means extracted features from the l-th layer of VGG19, and CX means the similarity between the features of z and $\hat{z}$.

Text Effect Consistency Loss L_{tec} is used to measure the L_1 distance between the normal text effect features and the augmented text effect features:

$$L_{tec} = \mathbb{E} \left\| E_x(X^*) - E_x(X) \right\|_1. \tag{6}$$

Adversarial Loss To make the generated images indistinguishable from the real, we apply an adversarial loss:

$$L_{adv} = \mathbb{E}[log(D(z))] + \mathbb{E}[log(1 - D(\hat{z}))]. \tag{7}$$

4. EXPERIMENTS

4.1. Datasets

We have done detailed statistics on the existing artistic font datasets, as shown in Table 1 and Fig.3. The existing datasets do not support style separability, which means the independent glyph and text effect reference cannot be provided. So we propose a new dataset, SSAF. In addition, we specifically modified the TextEffects [2] dataset.

SSAF[1]**:** We create a new style separable artistic font dataset, which contains 200 artistic fonts with combinations of glyphs and text effects. Among them, 100 are Chinese artistic fonts, each with 972 characters; 100 are English artistic fonts, each with 26 uppercase English letters. There are 99,800 images in total, in 320×320 resolution. In SSAF-CN, we firstly collect 10 categories of glyphs images. Then, we make some artistic effects and download some exquisite PSD effect files from the website, 10 in total. Finally, we use script files to synthesize all artistic font images, an overview in Fig.3(e)(f). And so is SSAF-EN.

TextEffects [2]: The TextEffects dataset contains 64 kinds of artistic fonts in 320×320 resolution. We use them as the target artistic font images, and we also add the source images and the glyph images corresponding to the artistic fonts. Their glyph style and type size are as consistent as possible with the artistic fonts.

4.2. Implementation Details

All of our encoders have six convolution layers, and the generator has six convolution and up-convolution layers. Each layer is equipped with instance normalization and ReLU. We use Adam optimizer with β_1 = 0.9 and β_2 = 0.999 for the whole DSE-Net, and train it 100 epochs with a learning rate of 0.0002. For all experiments, the batch size is set to 4 and $\lambda_{tec} = 100$, $\lambda_{CX} = 25$, $\lambda_1 = 100$, $\lambda_{adv} = 1$. The threshold in the cross-layer fusion mechanism h is set to 3. For each type of Chinese artistic font, 774 for training and 198 for testing. For each type of English artistic font, 20 for training and 6 for testing. In each process of forwarding propagation, the value n, m of the glyph reference and text effect reference inputs are set to 4. And these reference samples are randomly selected from the reference sample sets.

4.3. Performance Comparison

We compare the existing artistic font generation methods, such as TET-GAN [2], AGIS-Net [5], FET-GAN [4]. For a fair comparison, we use Heiti and Arial as the source images, which are commonly used in Chinese and English font generation tasks.

Visual Comparison. As shown in Fig.4, we first show three sets of challenging text effects (a)(b)(c), all of which have fine-grained textures. (d) is a set of English experimental results, and (e) is a set of results on the TextEffects dataset. We can observe that our method performs much better than other methods, especially in the completeness and continuity of the strokes. It can also obtain more precise and delicate textures when dealing with challenging text effects.

[1]https://github.com/moonlight03/DSE-Net

Table 1. Comparison of artistic fonts datasets.

Dataset	Size	Characters	Glyph Image	Style Separability
TextEffects [2]	320^2	CN,EN	without	nonsupport
CAGI [5]	64^2	CN	grayscale	nonsupport
Capitals64 [3]	64^2	EN	standard	nonsupport
SSAF(ours)	**320^2**	**CN,EN**	**standard**	**support**

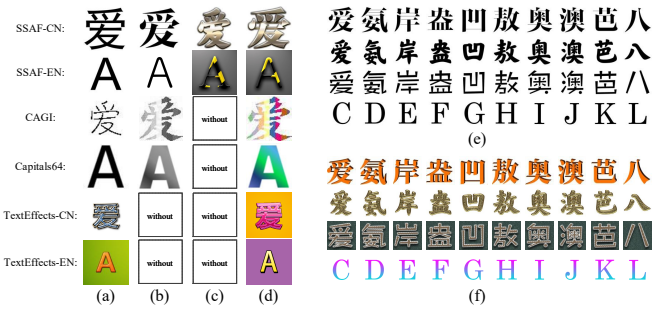


Fig. 3. Examples of the datasets. (a) Source images. (b) Glyph reference. (c) Text effect reference. (d) Artistic fonts. (e)(f) Glyph and artistic font examples in SSAF.

Quantitative Comparison. We further conduct quantitative comparison in terms of the L_1 and FID. As shown in Table 2, we compare the results of five sets of quantitative experiments, each of which is calculated from the entire artistic font test set. The best performance is shown in bold. And our method achieves the best performance with the lowest L_1 and FID in most cases.

4.4. Ablation Study

In this section, we demonstrate the effectiveness of the critical parts of our model through qualitative and quantitative comparison. All experiments in this section are conducted on 20 randomly selected artistic fonts.

1) Effectiveness of cross-layer fusion mechanism(CL). The CL provides content, glyph, and text effect information supplement in different phases of the up-sampling. Without CL, Fig.5(a) shows the generated result. We can see that the result cannot maintain the original content representation, and its appearance is chaotic.

In addition, we further explore the different characteristics of glyph style and text effect style in the high-level space and low-level space of the CNN. We use VGG19 to extract the features of 16,000 artistic font images, and calculate the

Table 2. Quantitative comparison of the DSE-Net and other methods.

Model	SSAF-Syjt09		SSAF-Kaiti10		SSAF-Songti06		SSAF-Encola02		TEF-Comicbook	
	FID ↓	L_1 loss ↓	FID ↓	L_1 loss ↓	FID ↓	L_1 loss ↓	FID ↓	L_1 loss ↓	FID ↓	L_1 loss ↓
TET-GAN [2]	109.38	0.2273	113.96	0.2170	131.39	0.2363	271.06	0.1031	325.04	0.2014
FET-GAN [4]	133.36	0.2189	142.23	0.2011	116.26	**0.2112**	269.97	0.1168	314.62	0.1776
AGIS-Net [5]	120.94	0.2199	108.13	0.2048	80.66	0.2187	257.69	0.1211	78.15	0.1224
DSE-Net	**107.81**	**0.2142**	**85.56**	**0.1963**	**66.64**	0.2174	**223.63**	**0.1028**	**75.27**	**0.1153**

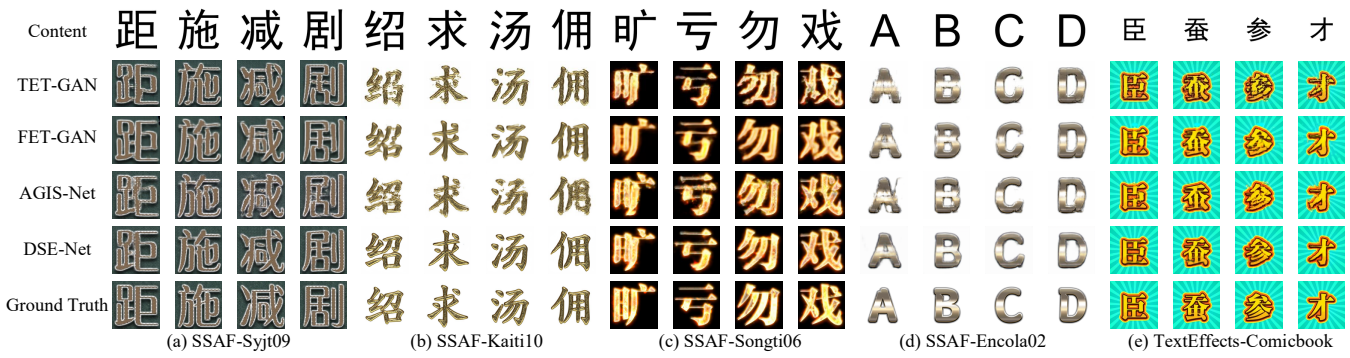
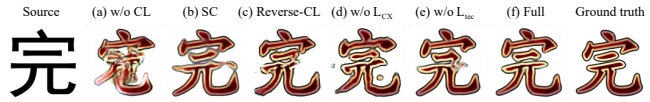


Fig. 4. Comparison with other methods (Zoom in for details). Experimental results are named after artistic fonts (i.e., SSAF-Syjt09, 'Syjt' is the target glyph and '09' is the index of the text effect). (a) Cloth pattern. (b) Metallic grain. (c) Flame. (d) English artistic fonts. (e) Artistic fonts in TextEffects dataset. Among them, the first three sets are challenging text effects.



	(a)	(b)	(c)	(d)	(e)	(f)	-
mL_1 loss↓	0.2413	0.1981	0.2054	0.2013	0.2004	**0.1957**	-
mFID↓	126.32	108.43	107.26	167.07	108.22	**106.63**	-

Fig. 5. Ablation study of the DSE-Net via quantitative and qualitative evaluation.

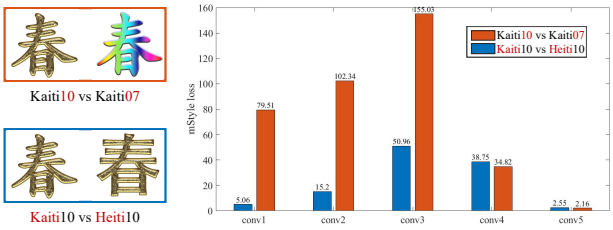


Fig. 6. Text effect difference (i.e., 10 vs 07) make the higher style loss at the low-level network and glyph difference(i.e., Kaiti vs Heiti) make the higher style loss at the high-level network.

mean style loss [7] between different categories of artistic font. One set of comparisons is shown in Fig.6. By comparing a large number of style loss produced by each layer of convolution, we can conclude that the text effects have received more attention on the low-level space, and the glyphs have received more attention on the high-level space. In order to verify this conclusion, we modify the CL and let the generator use high-level text effect features and low-level glyph features to produce the Reverse-CL. Moreover, we also compare the CL with commonly used skip-connection(SC) [20], which is often adopted to transfer feature maps from encoder to decoder at each layer. Fig.5(b)(c)(f) prove the superiority of the CL.

2) Effectiveness of contextual loss. The contextual loss leads the model to pay more attention to higher-dimensional features, not pixels. Fig.5(d)(f) show that L_{CX} effectively improves the quality of the generated result, such as removing noise and making the lines smooth.

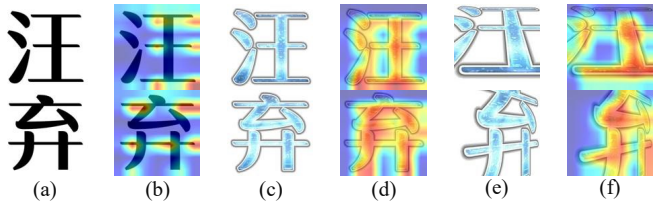
3) Effectiveness of text effect consistency loss. Before we apply the L_{tec}, we have verified an objective problem: inconsistent glyph information between the text effect reference and target artistic fonts will bring different degrees of disturbance for final results. In Table 3, our target glyph is Songti. The Songti text effect reference achieves the best results, but other glyphs bring different degrees of disturbance. After we apply the L_{tec}, Fig.5(e)(f) show that the L_{tec} improves the continuity and consistency of strokes and achieves the best results in quantitative experiments.

4.5. Visualization

We use Grad-CAM [21] to visualize the glyphs and text effects heat-maps. As shown in Fig.7, the glyph encoder will

Table 3. Inconsistent glyph information between the text effect reference and the target artistic font will disturb the results. The first row represents the different glyphs in the text effect reference, where Songti is the target glyph.

	Songti	Heiti	Kaiti	Syjt	Cola
L_1 loss↓	**0.1689**	0.1711	0.1720	0.1705	0.1708
FID↓	**69.11**	72.51	70.14	71.69	71.56



(a) (b) (c) (d) (e) (f)

Fig. 7. Visualization of the attention maps: (a) Glyph images, (b) Grad-CAM heat-maps of the glyph, (c) Artistic font images, (d) Grad-CAM heat-maps of the text effect, (e) Perspective transformation examples, (f) Grad-CAM heat-maps of the perspective transformation text effect.

pay attention to the local characteristics of the glyph, such as the thickness of the stroke, the ends, and the structure. For the text effect, the text effect encoder will pay attention to some bright colors and textures.

5. CONCLUSION

We propose a new idea for synthesizing artistic fonts, disentangled style encoding. Based on this idea, we propose a novel model, termed DSE-Net, which can independently extract glyph and text effect features to generate fine-grained artistic fonts. Meanwhile, we create a new dataset, SSAF, which is the first artistic font dataset that supports style separability. Extensive experiments demonstrate that our model is capable of generating artistic fonts with complex glyphs and text effects.

6. ACKNOWLEDGEMENT

This work is supported in part by the Oversea Innovation Team Project of the "20 Regulations for New Universities" funding program of Jinan (Grant no. 2021GXRC073)

7. REFERENCES

[1] Shuai Yang, Jiaying Liu, and Zhouhui Lian et al., "Awesome typography: Statistics-based text effects transfer," in *CVPR*, 2017, pp. 7464–7473.

[2] Shuai Yang, Jiaying Liu, and Wenjing Wang et al., "Tet-gan: Text effects transfer via stylization and destylization," in *AAAI*, 2019, pp. 1238–1245.

[3] Samaneh Azadi, Matthew Fisher, and Vladimir G Kim et al., "Multi-content gan for few-shot font style transfer," in *CVPR*, 2018, pp. 7564–7573.

[4] Wei Li, Yongxing He, and Yanwei Qiet al., "Fet-gan: Font and effect transfer via k-shot adaptive instance normalization," in *AAAI*, 2020, pp. 1717–1724.

[5] Yue Gao, Yuan Guo, and Zhouhui Lian et al., "Artistic glyph image synthesis via one-stage few-shot learning," in *TOG*, 2019, pp. 1–12.

[6] Leon A Gatys, Alexander S Ecker, and Matthias Bethge, "A neural algorithm of artistic style," *arXiv preprint arXiv:1508.06576*, 2015.

[7] Leon A Gatys, Alexander S Ecker, and Matthias Bethge et al., "Image style transfer using convolutional neural networks," in *CVPR*, 2016, pp. 2414–2423.

[8] Justin Johnson, Alexandre Alahi, and Li Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *ECCV*, 2016, pp. 694–711.

[9] Tero Karras, Samuli Laine, and Timo Aila, "A style-based generator architecture for generative adversarial networks," in *CVPR*, 2019, pp. 4401–4410.

[10] Tianwei Lin, Zhuoqi Ma, and Fu Li et al., "Drafting and revision: Laplacian pyramid network for fast high-quality artistic style transfer," in *CVPR*, 2021, pp. 5141–5150.

[11] Song Park, Sanghyuk Chun, and Junbum Cha et al., "Few-shot font generation with localized style representations and factorization," *arXiv preprint arxiv:2009.11042*, 2020.

[12] Junbum Cha, Sanghyuk Chun, and Gayoung Lee et al., "Few-shot compositional font generation with dual memory," in *ECCV*, 2020, pp. 735–751.

[13] Song Park, Sanghyuk Chun, and Junbum Cha et al., "Multiple heads are better than one: Few-shot font generation with multiple localized experts," in *ICCV*, 2021.

[14] Yiming Gao et al., "Gan-based unpaired chinese character image translation via skeleton transformation and stroke rendering," in *AAAI*, 2020, pp. 646–653.

[15] Yankun Xi, Guoli Yan, Jing Hua, and Zichun Zhong, "Jointfontgan: Joint geometry-content gan for font generation via few-shot learning," in *MM*, 2020, pp. 4309–4317.

[16] Sharon Fogel, Hadar Averbuch-Elor, and Sarel Cohen et al., "Scrabblegan: Semi-supervised varying length handwritten text generation," in *CVPR*, 2020, pp. 4324–4333.

[17] Anna Zhu, Xiongbo Lu, and Xiang Bai et al., "Few-shot text style transfer via deep feature similarity," in *TIP*, 2020, vol. 29, pp. 6932–6946.

[18] Anna Zhu, Qiyang Zhang, and Xiongbo Lu et al., "Character image synthesis based on selected content and referenced style embedding," in *ICME*, 2019, pp. 374–379.

[19] Roey Mechrez, Itamar Talmi, and Lihi Zelnik-Manor, "The contextual loss for image transformation with non-aligned data," in *ECCV*, 2018, pp. 768–783.

[20] Olaf Ronneberger, Philipp Fischer, and Thomas Brox, "U-net: Convolutional networks for biomedical image segmentation," in *MICCAI*, 2015, pp. 234–241.

[21] Ramprasaath R Selvaraju et al., "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *ICCV*, 2017, pp. 618–626.