REPORT ON SHAPE-MATCHING GAN ALGORITHM

Xinyue Chen
New York University

Binfang Ye
New York University

1 INTRODUCTION

There has recently been a surge in the topic of Neural style transfer in computer vision. Among them, style transfer images or Artistic text have been widely used in advertisement and entertainment applications. Some previous literature focused on the scale of textures like the texture strength, or the size of texture patterns, but overlooked crucial elements such as glyph deformation. A major challenge is that the stylistic degree in terms of shape deformation is uncontrollable. There has been no work about the real-time control of glyph deformations. In addition, real-time rendering text with different deformation degree requires large text dataset with corresponding deformation levels. We consider a real-time text style transfer algorithm Shape-Matching GAN (Yang et al., 2019), which utilizes the novel bidirectional shape matching framework to address the above challenges.

In our project we conducted 3 sets of experiments on (i) validating the the result in the Shape Matching GAN paper (ii) ablating the effect of certain components of the algorithms (iii) applying the algorithm to real-world images, namely, *easy-drawings*.

2 RELATED WORK

Neural Style Transfer is proposed by Gatys et al. (2016), in which they leveraged the powerful representation ability of neural networks to transfer styles formulated as the Gram matrix of deep feature. Johnson et al. (2016) employed a feed-forward StyleNet using the loss of Neural Style Transfer (Gatys et al., 2016) for fast style transfer. Following literature exploited Generative Adversarial Network (GAN) (Mirza & Osindero, 2014) to transfer specialized styles such as cartoons, paintings and makeups.

Artistic Text Style Transfer was first raised by Yang et al. (2017). A patch-based model UT-Effect (Yang et al., 2018) stylized the text with arbitrary textures and achieved glyph deformations by shape synthesis, which shows promise for more application scenarios. Meanwhile, Jing et al. (2018) proposed a stroke-controllable neural style transfer network (SC-NST) with adaptive receptive fields for stroke size control. Compared to prior methods, the Shape-Matching GAN algorithm enables the continuous adjustment of glyph deformation degree via a controllable parameter, thereby achieving *Multi-scale and Real-time style control*.

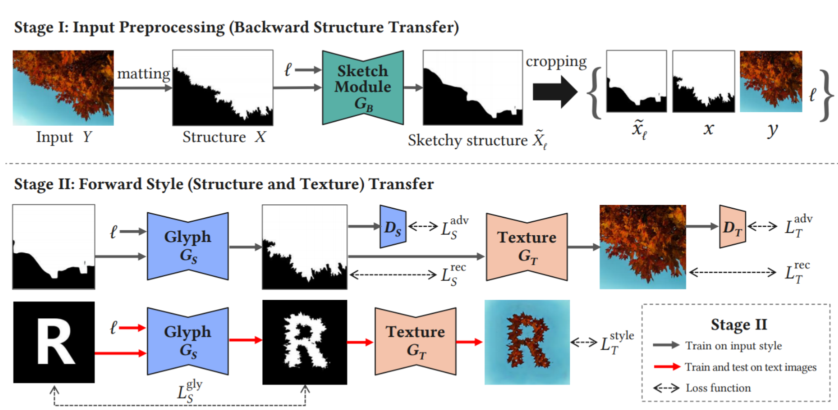


Figure 1: Overview of bidirectional shape matching framework

3 Approach

3.1 Shape-Matching GAN

Bidirectional shape matching GAN is an end-to-end algorithm that transfers the style of the source image to the text picture. The algorithm can generate diverse, controllable, and high-quality stylized text. The step of implementing this algorithm includes two main parts, backward transfer, and forward transfer. The figure 1 shows an overview of the bidirectional shape matching framework.

3.1.1 Backward Structure Transfer

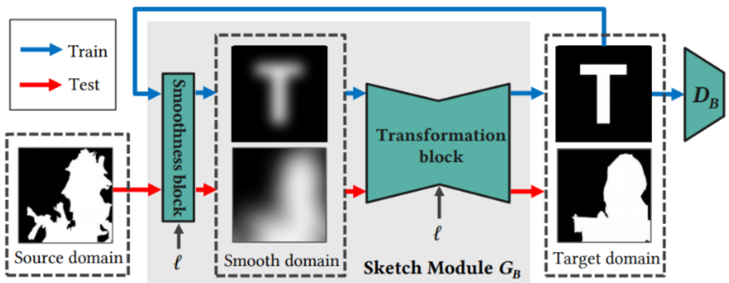


Figure 2: Overview of Backward Structure Transfer

The backward transfer is able to learn the characteristics from text images and transfer them to the structure picture (i.e. edited the source image by PhotoShop or applied Matting algorithm to the source image) at different coarse levels (i.e. controlled by deformation degree l). The algorithm applied Gaussian kernel, whose standard deviation $\sigma = f(l)$, where $f(l) = 16l + 8$, to control the coarse levels. The key idea is to use a smoothness block (i.e. Gaussian kernel) to eliminate the details and obtain the sketchy contour. In the meantime, it can map the text images and structure picture to the same smooth domain. Therefore, the deformation level becomes controllable. To learn the glyph characteristics, the transformation block will map the smoothed text images back to the original text domain to gain the sketchy structure images containing glyph features at different deformation levels. In the backward transfer process, it has a generator to produce more text-like contours that can help the algorithm to learn the glyph features. In addition, there is a discriminator that will decide the quality of the images generated by the generator. After processed backward transfer, artistry is met since the output contains the shape information of the style reference. The figure 2 shows the overview of backward structure transfer.

3.1.2 Forward Transfer

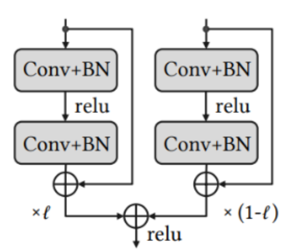


Figure 3: Controllable ResBlock

Structure Transfer

Given the outputs (i.e. structure pictures with different deformations) from Backward structure transfer, forward structure transfer (glyph network) will map the outputs to the original structure picture so that the network can learn the shape features of the structure picture that will be transferred to the text images. However, mapping the outputs to the structure picture is a many-to-one process that will crash the model since it will lead to similar results regardless of the deformations. To address this problem, forward structure transfer employs data augmentation and Controllable ResBlock techniques. Firstly, the structure picture and the outputs will be randomly paired and cropped into 256×256 sub-images for training. Secondly, Yang et al. (2019) modify the architecture of StyleNet designed by Johnson et al. (2016). Yang et al. replace a single ResBlock with a linear combination of two ResBlocks weighted by l.(i.e. Controllable ResBlock). Using the Controllable ResBlock can avoid the many-to-one issue and compromises the greatest(tiniest) shape deformation. In some situations, the larger the value l, the harder it to recognize the text. Therefore, for forward structure transfer loss, Yang et al. (2019) add L_S^{gly} in the loss function to avoid the illegibility issue of outputs.

Texture Transfer

Given the outputs from the forward structure transfer, texture transfer is easy to achieve via existing algorithms such as the greedy-based Image Analogy and the optimization-based Neural Doodle. The algorithm is able to transfer the texture from source style to the output. Eventually, it generates source stylized text images.

3.2 Implementation Details

We used the authors' open-source code on Shape Matching GAN (Yang et al., 2019). We employed the default hyper-parameters as suggested by the authors. We also included the optional glyph preserve loss to improve the controllability of glyph deformation. For training the networks of Shape Matching GAN, we used the GTX 1080 Ti / 2080 Ti GPU nodes of NYU Shanghai High Performance Computing cluster. For some other image processing or testing procedures, we used the computation resources of Google Colab. Our reproduction results on the author's dataset is shown in figure 4.

(a) fire texture (b) sakura texture (c) smoke texture

Figure 4: Style transfer results on the Chinese character, English letter, and Arabic numeral.

4 Experiments

4.1 Ablation studies

In their paper, the authors already did ablation study on 3 components of the Shape Matching GAN algorithm: the Controllable ResBlock (CR), the texture network (TN) and the Transformation Block (TB).

Additionally, we consider 2 other components to ablate on. *Image augmentation* is applied to both the structure transfer and the texture transfer steps. An optional *Glyph preserve loss* could be included in the forward structure transfer, and is claimed to "force" the structure transfer result to maintain the main stroke part of the input.

w/o Image augmentation We consider taking out the image augmentation step for training G_S and G_T. Results in figure 5 show that without image augmentation, the algorithm is less consistent with the reference style. This is potentially due to over-fitting on glyphs and their styles in the train set, and fail to capture the features of glyphs in the test set. This problem is well-handled by the image augmentation steps, which maintains shape and style consistency even with a large degree l of deformation.

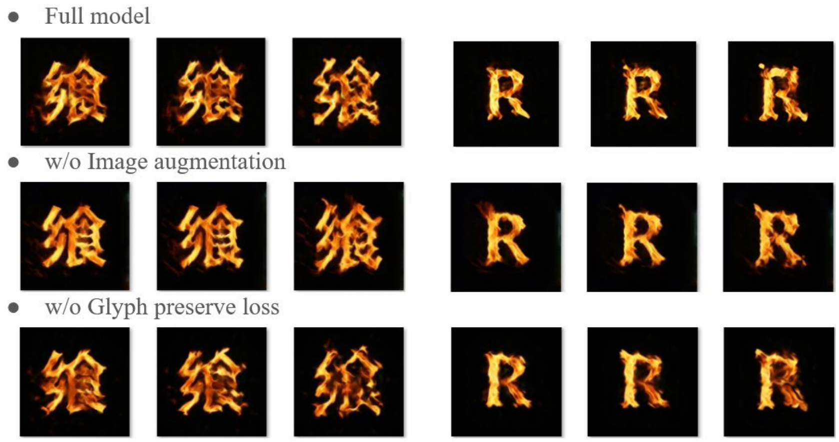


Figure 5: Example results of our ablation study on image augmentation and glyph preserve loss. We choose Chinese characters and English letters with relatively high complexity (which should be considered harder in learning the style transfer). Note that deformation degree increases from left to right.

w/o Glyph preserve loss. We consider not including the optional glyph preserve loss in the training objective of G_S. In this case, the algorithm constantly fail to preserve the the main stroke part of the input glyph, as illustrated in in figure 5. When the deformation degree is high, it even fails to preserve the basic structure of the glyph. Therefore, the glyph preserve loss is essential for balancing between the legibility and artistry for text style transfer task.

4.2 Application on easy-drawings

4.2.1 Data generation

To produce an easy-drawing dataset (30 training data + 7 test data), we use PhotoShop to edit easy-drawing pictures with a gradient background. Before we edit the easy-drawing pictures downloaded from Google, we apply the mean-shift algorithm on them since we believe it is able to filter some noises from the downloaded images. Then we change the stroke color to red. However, the stroke width is not the same as the training set given by the authors. This is considered a challenge for generalizing the learnt representations from the text data to the easy-drawing data. The figure 6 shows the process of making an easy-drawing data point.

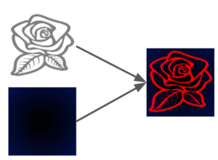


Figure 6: Process of making easy-drawing data

4.2.2 Training and results

In this experiment we consider different distributions of the training set, to study which combination gives a better performance.

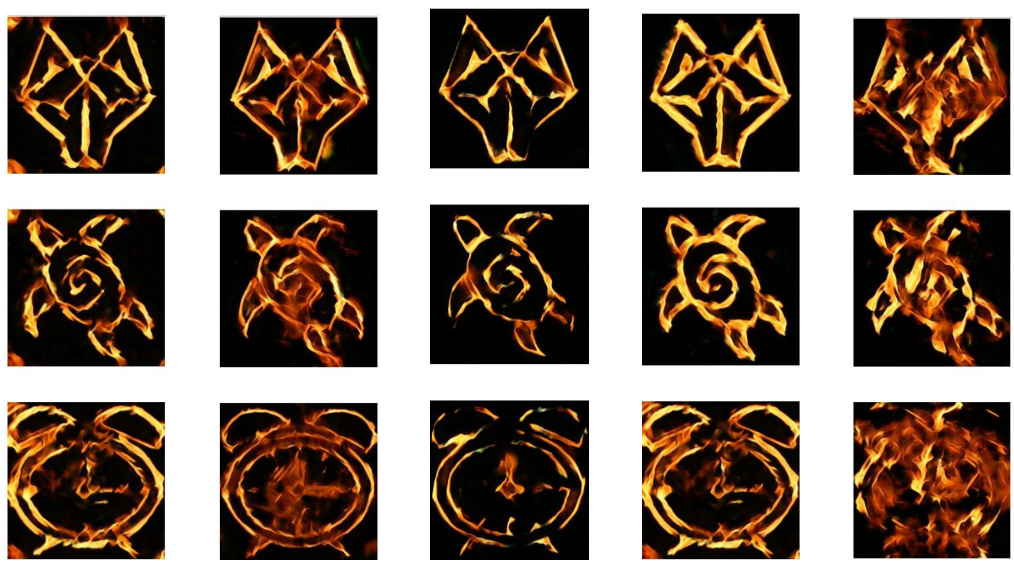
text-only. 708 text images.

mixture-less balanced. 678 text images + 30 easy-drawing images.

mixture-more balanced. 70 text images + 30 easy-drawing images.

drawing-only. 30 easy-drawing images.

Due to limitation of computational resources, we only tune the batch size $b \in 16, 32$ and learning rate $lr \in 0.001, 0.0001$, from which we choose the most high-performing combination. Results are presented in figure 7. Some observations are (i) Even without learning on the easy-drawing images (**text-only**), the algorithm already has a fair performance on style transfer of the easy-drawings. But there are some flaws in the background, indicating that the GAN is not capturing the glyph very well. (ii) All categories that include easy-drawing images as train data succeeded in learning the style transfer for easy-drawings. When the train data contains both text and easy-drawing images (**mixture** type), the algorithm appears to learn better features of the style. Also, it is able to preserve the difference of stroke width between the 2 data sources. These indicate the potential advantage of leveraging the features of text style transfer in the style transfer of easy-drawings. (iii) Some issues in the results are: As the stoke width of some easy-drawings are pretty small, the model sometimes disconnect the drawing after style transfer; When the deformation degree $l = 1$ (largest), the model produces illegible results for most test samples of easy-drawings.



(a) text-only (b) mixture-less balanced (c) mixture-more balanced. (d) drawing-only (e) extreme deformation

Figure 7: Style transfer results on the easy-drawings. All results in $(a) \sim (d)$ uses deformation degree of $l = 0.4$. For (e), we show some of the results with $l = 1$ to illustrate the illegibility issue in extreme case of deformation

5 Conclusion and Future Directions

In this project we reproduce and study the extensions of the Shape Matching-GAN algorithm. By analyzing and ablating on the components of the algorithm, we get to know which designs are effective for learning text style transfer. Furthermore, we generated data from a new source of *easy-drawing*. We found that training on both text image and easy-drawing images improves the generalization properties of the algorithm.

To address the existing problems discussed in section 4.2.2, we consider generating more data from a boarder range of drawings. This would further help the model to learn features of image style transfer. However, we believe including the text data would be helpful since collection of drawing data could be expensive. Moreover, Shape-Matching GAN algorithm does not performs well on images with small stroke width. So we consider adapting stroke-controllable neural style transfer network(SC-NST) to address the issue.

REFERENCES

Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2414–2423, 2016.

Yongcheng Jing, Yang Liu, Yezhou Yang, Zunlei Feng, Yizhou Yu, Dacheng Tao, and Mingli Song. Stroke controllable fast style transfer with adaptive receptive fields. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 238–254, 2018.

Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, pp. 694–711. Springer, 2016.

Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

Shuai Yang, Jiaying Liu, Zhouhui Lian, and Zongming Guo. Awesome typography: Statistics-based text effects transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7464–7473, 2017.

Shuai Yang, Jiaying Liu, Wenhan Yang, and Zongming Guo. Context-aware text-based binary image stylization and synthesis. *IEEE Transactions on Image Processing*, 28(2):952–964, 2018.

Shuai Yang, Zhangyang Wang, Zhaowen Wang, Ning Xu, Jiaying Liu, and Zongming Guo. Controllable artistic text style transfer via shape-matching gan. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4442–4451, 2019.