SwapText: Image Based Texts Transfer in Scenes

Qiangpeng Yang, Jun Huang, Wei Lin
Alibaba Group
yqp0424@gmail.com, huangjun.hj@alibaba-inc.com, weilin.lw@alibaba-inc.com

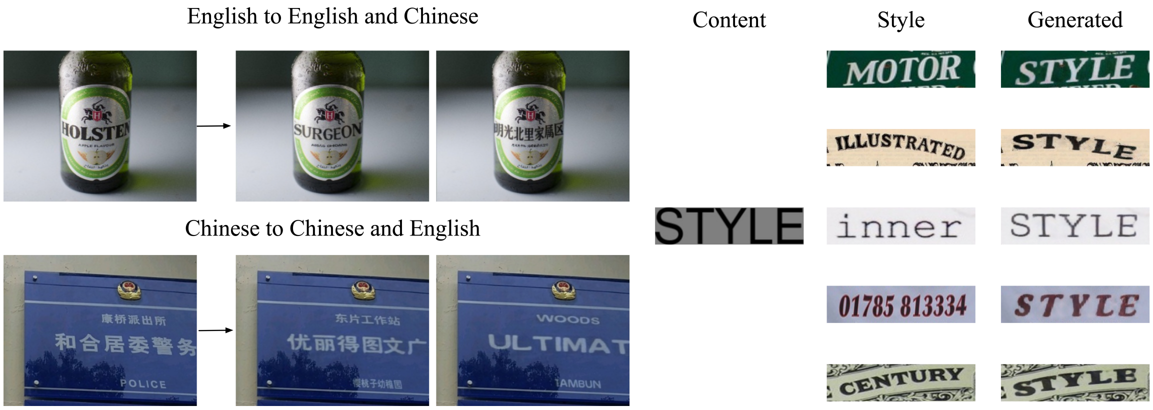


Figure 1. Arbitrary text style transfer in scene text images. (*Left*) Our model learns to perform diverse translation between multilanguage. (*Right*) Style-guide transfer

Abstract

Swapping text in scene images while preserving original fonts, colors, sizes and background textures is a challenging task due to the complex interplay between different factors. In this work, we present SwapText, a three-stage framework to transfer texts across scene images. First, a novel text swapping network is proposed to replace text labels only in the foreground image. Second, a background completion network is learned to reconstruct background images. Finally, the generated foreground image and background image are used to generate the word image by the fusion network. Using the proposing framework, we can manipulate the texts of the input images even with severe geometric distortion. Qualitative and quantitative results are presented on several scene text datasets, including regular and irregular text datasets. We conducted extensive experiments to prove the usefulness of our method such as image based text translation, text image synthesis, etc.

1. Introduction

Imagine being able to swap text in scene images while keeping the original fonts, colors, sizes and background textures within seconds, and without hours of image editing. In this work, we aim to realize this goal with an algorithm that automatically replaces the text in scene images. The core challenge of text swapping lies in generating visually realistic text and keeping coherent style with the original text.

Text swapping or text replacement is relevant in many scenarios including text detection, text recognition, text transfer in posters and other creative applications. For text detection and recognition tasks, text swapping is a very useful data augmentation approach. Witness the great success of deep neural networks (DNN) in various computer vision tasks, obtaining large amounts of annotated training images has become the bottleneck for training DNN models. The easiest and most widely used methods augment training images by geometric transformation, such as translation, rotation and flipping, *etc*. Recently, image synthesis based approaches [11, 7, 39] have been proposed for training text detection and recognition models. These approaches create new images from text-free images by modeling the physical behaviors of light and energy in combination of different rendering techniques. However, the synthetic images do not fully cohere with the images in scenes, which is critically important while applying the synthesized images to train DNN models.

In most recent years, many image generation models, such as generative adversarial networks (GANs) [6] , variational autoencoders (VAE) [17], and autogressive models [25] have provided powerful tools for realistic image generation tasks. In [9, 38, 33], GANs are used for image completion that generates visually realistic and semantically plausible pixels for the missing regions. [21, 8, 28, 22] have exploited these networks to generate novel person images with different poses or garments.

Based on GANs, we present a unified framework SwapText for text swapping in scenes. A few examples can be seen in Figure 1. We adopt a divide-and-conquer strategy, decompose the problem into three sub-networks, namely text swapping network, background completion network and the fusion network. In the text swapping network, the features of content image and style image are extracted simultaneously and then combined by a self-attention network. To better learn the representation of content image, we use a Content Shape Transformation Network (CSTN) to transform the content image according to the geometrical attributes of the style image. According to our experiments, this transformation process has significantly improved image generation, especially for perspective and curved images. Then, a background completion network is used to generate the background image of style image. Because we need to erase the original text stroke pixels in the style image and fill with appropriate texture according to the content image. Finally, the output of text swapping network and background completion network are fed into the fusion network to generate more realistic and semantically coherent images. The whole framework are end-to-end trainable, and extensive experiments on several public benchmarks demonstrate its superiority in terms of both effectiveness and efficiency.

Our contributions are summarized as follows:

- We design an end-to-end framework namely SwapText, which contains three sub-networks, text swapping network, background completion network and the fusion network.
- We propose a novel text swapping network that replace the text in the scene text images, while keeping the original style.
- We demonstrate the effectiveness of our method for scene text swapping with high-quality visual results, and also show its application to text image synthesis, image based text translation, *etc*.

2. Related Work

Text Image Synthesis Image synthesis has been studied extensively in computer graphics research [4]. Text image synthesis is investigated as a data augmentation approach for training accurate and robust DNN models. For example, Jaderberg *et al.* [11] use a word generator to generate synthetic word images for text recognition task. Gupta *et al.* [7] develop a robust engine to generate synthetic text image for both text detection and recognition tasks. The target of text image synthesis is to insert texts at semantically sensible regions within the background image. Many factors affect the true likeness of the synthesized text images, such as text size, text perspective, environment lighting, *etc*. In [39], Zhan *et al.* achieve verisimilar text image synthesis by combining three designs including semantic coherence, visual attention, and adaptive text appearance. Although the text images synthesis are visually realistic, there are many differences between synthetic images and real images. For instance, comparing to the real images the fonts of text and background image in synthetic images are very limited.

In most recently, GAN based image synthesis technology has been further explored. In [41], Zhan *et al.* present an spatial fusion GAN that combines a geometry synthesizer and an appearance synthesizer to achieve synthesis realism in both geometry and appearance spaces. Yang *et al.* [36] use bidirectional shape matching framework control the crucial stylistic degree of the glyph through an adjustable parameter. GA-DAN [40] present an interesting work that is capable of modelling cross domain shifts concurrently in both geometry space and appearance space. In [2], MC-GAN is proposed for font style transfer in the set of letters from A to Z. Wu *et al.* [34] propose an end-to-end trainable style retention network to edit text in natural images.

Image Generation With the great success of generative models, such as GANs [6], VAEs [17] and autogressive models [25], realistic and sharp image generation has attracted more and more attention lately. Traditional generative models use GANs [6] or VAEs [17] to map a distribution generated by noise z to the distribution of real data. For example, GANs [6] are used to generate realistic faces [37, 3, 15] and birds [29].

To control the generated results, Mirza *et al.* [23] proposed conditional GANs. They generate MNIST digits conditioned on class labels. In [12], karacan *et al.* generate realistic outdoor scene images based on the semantic layout and scene attributes, such as day-night, sunny-foggy. Lassner *et al.* [19] generated full-body images of persons in clothing based on fine-grained body and clothing segments. The full model can be conditioned on pose, shape, or color. Ma *et al.* [21, 22] generate person images based on images and poses. Fast face-swap is proposed in [18] to transform an input identity to a target identity while preserving pose, facial expression and lighting.

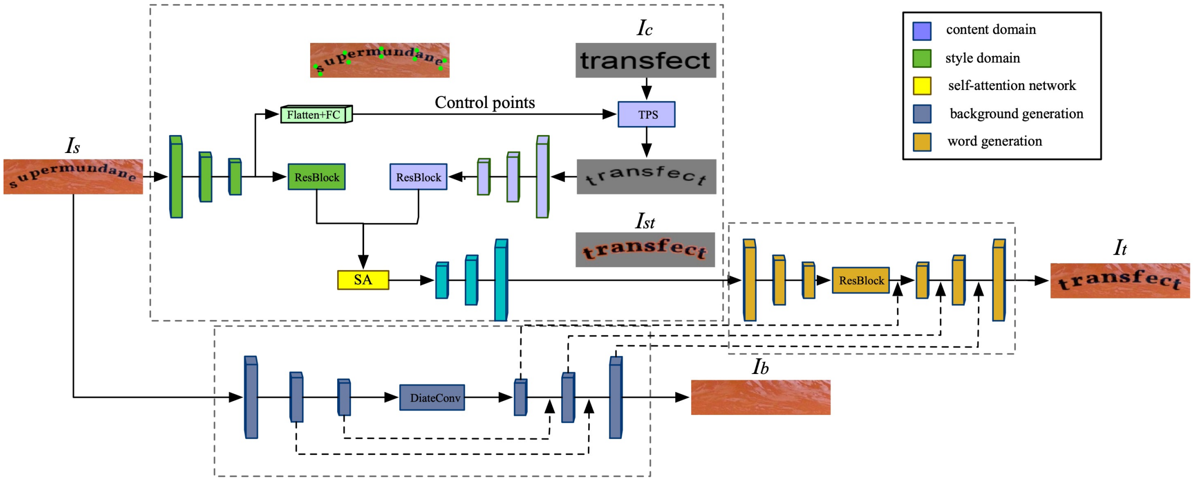


Figure 2. The framework of our proposed method. It contains three sub-networks: text swapping network, background completion and fusion network.

Image Completion Recently, GAN-based approaches have emerged as a promising paradigm for image completion. Iizuka *et al*. [9] proposed to use global and local discriminators as adversarial losses, where both global and local consistency are enforced. Yu *et al*. [38] use a contextual attention layer to explicitly attend on related feature patches at distant spatial locations. Wang *et al*. [33] use a multi-column network to generate different image component in a parallel manner, and an implicit diversified MRF regularization is adopted to enhance local details.

3. Methodology

Given a scene text image $I_s \in \mathbb{R}^{H\times W\times 3}$, our goal is to replace the text based on a content Image $I_c \in \mathbb{R}^{H\times W\times 3}$ while keeping the original style. As illustrated in Figure 2, our framework consists of text swapping network, background completion network and the fusion network. The text swapping network firstly extracts the style features from I_s and content features from I_c, then combine these two features by a self-attention network. To learn a better representation of content, we use a Content Shape Transformation Network (CSTN) to transform the content image I_c according to the geometrical attributes of the style image I_s. The background completion network is used to reconstruct the original background images I_b of style image I_s. Finally, the outputs of text swapping network and background completion network are fused by the fusion network to generate the final text images.

3.1. Text Swapping Network

Text instances in real-world scenarios have diverse shapes, *e.g*., in horizontal, oriented or curved forms. The main purpose of text swapping network is to replace the content of the style image I_s, while keeping the original style, especially text shapes. To improve the performance of irregular text image generation, we propose a Content Shape Transformation Network (CSTN) to map the content image into the same geometric shape of the style image. Then the style image and transformed content image are encoded by 3 down-sampling convolutional layers and several residual blocks. To combine the style and content features adequately, we feed them into a self-attention network. For decoding, 3 up-sampling deconvolutional layers are used to generate the foreground images I_f.

3.1.1 Content Shape Transformation Network

The definition of text shape is critical for content shape transformation. Inspired by the text shape definition in text detection [20] and text recognition [35] field, the geometrical attributes of text can be defined with $2K$ fiducial points $P = \{p_1, p_2, ..., p_{2K}\}$, which is illustrated in Figure 3.

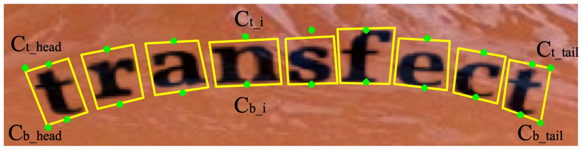


Figure 3. Illustration of text shape definition.

A text instance can be viewed as an ordered character sequence $T = \{C_1, ..., C_i, ..., C_n\}$, where n is the number of characters. Each character C_i has a bounding box B_i, which is annotated with a free-form quadrilateral. First, we construct two center point lists $C_{top} = \{C_{t\_head}, C_{t\_1}, ..., C_{t\_n}, C_{t\_tail}\}$ and $C_{bottom} = \{C_{b\_head}, C_{b\_1}, ..., C_{b\_n}, C_{b\_tail}\}$, which contains the top center and bottom center for each B_i. Then we evenly spaced sampling K fiducial points in C_{top} and C_{bottom}. For the points not in C_{top} or C_{bottom}, the values are linearly interpolated with two nearest center points. In this way, the shape of the text instance is precisely described by the fiducial points. In our experiments, K is set to 5.

To yield the text shape of input style image, we employ a lightweight predictor which shares the down-sampling convolutional layers with style image encoder, as illustrated in Figure 2. The output of this predictor is $\hat{P} = \{\hat{p}_1, \hat{p}_2, ...\hat{p}_{2K}\}$, which represents the geometrical attributes of the input image. We adopt $smooth_{L_1}$ loss the loss function of this predictor,

$$\mathcal{L}_P = \frac{1}{2K} \sum_{i=1}^{2K} smooth_{L_1}(p_i - \hat{p}_i), \tag{1}$$

Given the geometrical attributes of style image, we transform the content image through the Thin-Plate-Spline (TPS) module. The transform process is shown in Figure 4.

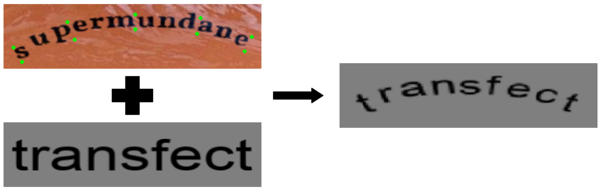


Figure 4. The shape transform process of content image.

3.1.2 Self-attention Network

After encoding the content and style images, we feed both feature maps to a self-attention network that automatically learns the correspondences between the content feature map F_c and style feature map F_s. The output feature map is F_{cs}, and the architect of self-attention network is presented in Figure 5 (a)

The content feature F_c and style feature F_s are firstly concatenated along their depth axis. Then we follow the similar self-attention mechanism in [42] to produce the output feature map F_{cs}.

We adopt the L_1 loss to as our text swapping network loss function, which is as follows,

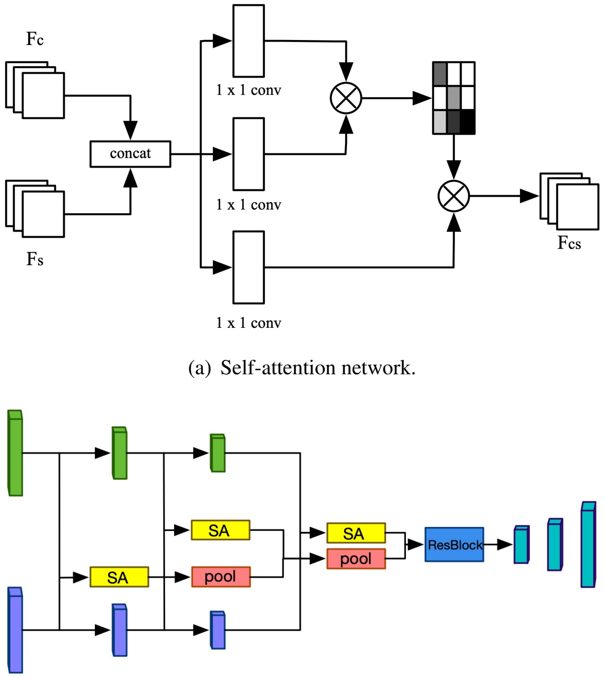
$$\mathcal{L}_{swap} = \left\| G_{swap}(I_s, I_t) - I_{st} \right\|_1, \tag{2}$$

where G_{swap} denotes the text swapping network, and I_{st} is the ground truth of text swapping network.

In addition to this single-level stylization, we further develop a multi-level stylization pipeline, as depicted in Figure 5 (b). We apply self-attention network sequentially to multiple feature layers to generate more realistic images.

3.2. Background Completion Network

The text swapping network mainly focus on the foreground image generation, while the background images also play a important role of the final image generation. To generate more realistic word image, we use a background completion network to reconstruct the background image, whose architecture is illustrated in Table 1. Most existing image completion approaches fill the pixels of the image by borrowing or copying textures from surrounding regions. The general architecture follows an encoder-decoder structure, we use dilated convolutional layer after the encoder to compute the output pixel with larger input area, By using dilated convolutions at lower resolutions, the model can effectively ”see” a larger area of the input image.



(a) Self-attention network.

(b) Multi-level stylization.

Figure 5. Architecture of self-attention network. (a) Self-attention network. (b) Multi-level self-attention.

The background completion network is optimized with both $\mathcal{L}_1$ loss and GAN loss. We use G_b and D_b to denote the background generator and discriminator, the overall loss for background generation are as follows,

$$\begin{aligned}\mathcal{L}_B = \mathbb{E}[\log D_b(I_b, I_s) + \log(1 - D_b(\hat{I}_b, I_s))] + \\ \lambda_b \left\| I_b - \hat{I}_b \right\|_1,\end{aligned} \tag{3}$$

where I_b and $\hat{I}_b$ are ground truth and predicted background images. λ_b is the balance factor and is set to 10 in our experiments.

3.3. Fusion Network

In this stage, the output of text swapping network and background completion network are fused to generate the complete text images. As the pipeline illustrated in Fig-

Table 1. Architecture of background completion network.

Type	Kernel	Dilation	Stride	Channels
conv	5×5	1	1×1	32
conv	3×3	1	2×2	64
conv	3×3	1	1×1	64
conv	3×3	1	1×1	64
conv	3×3	1	2×2	128
conv	3×3	1	1×1	128
conv	3×3	1	1×1	128
conv	3×3	1	2×2	256
conv	3×3	1	1×1	256
conv	3×3	1	1×1	256
diated conv	3×3	2	1×1	256
diated conv	3×3	4	1×1	256
diated conv	3×3	8	1×1	256
deconv	3×3	1	$\frac{1}{2} \times \frac{1}{2}$	256
conv	3×3	1	1×1	256
conv	3×3	1	1×1	256
deconv	3×3	1	$\frac{1}{2} \times \frac{1}{2}$	128
conv	3×3	1	1×1	128
conv	3×3	1	1×1	128
deconv	3×3	1	$\frac{1}{2} \times \frac{1}{2}$	64
conv	3×3	1	1×1	64
conv	3×3	1	1×1	64
output	3×3	1	1×1	3

ure 2, the fusion network follow an encoder-decoder architecture. Similar to [34], we connect the decoding feature maps of the background completion network to the corresponding feature maps with the same resolution in the upsampling phase of the fusion decoder. We use G_{fuse} and D_{fuse} to denote the generator and discriminator network respectively. The loss function of fusion network can be formulated as follows,

$$\begin{aligned} \mathcal{L}_F = \mathbb{E}[\log D_{fuse}(I_t, I_c) + \log(1 - D_{fuse}(\hat{I}_t, I_c))] + \\ \lambda_{fuse}\left\|I_t - \hat{I}_t\right\|_1 , \end{aligned} \tag{4}$$

where $\hat{I}_t$ is the output of the generator and λ_{fuse} is the balance factor which is set to 10 in our experiment.

In order to make more realistic images, we also introduce VGG-loss to the fusion module following the similar idea of style transfer network [5, 26]. There two parts of VGG-loss, the perceptual loss and style loss, as follows,

$$\begin{aligned} \mathcal{L}_{vgg} &= \lambda_1 \mathcal{L}_{per} + \lambda_2 \mathcal{L}_{style} \\ \mathcal{L}_{per} &= \mathbb{E}[\sum_i \left\|\phi_i(I_t) - \phi_i(\hat{I}_t)\right\|_1] \\ \mathcal{L}_{style} &= \mathbb{E}_j[\left\|G_j^{\phi}(I_t) - G_j^{\phi}(\hat{I}_t)\right\|_1], \end{aligned} \tag{5}$$

where ϕ_i is the activation map from $relu1\_1$ to $relu5\_1$ layer of VGG-19 model. G is the Gram matrix. λ_1 and λ_2 are the balance factors respectively.

The loss function of the whole framework is:

$$\mathcal{L} = \mathcal{L}_P + \mathcal{L}_{swap} + \mathcal{L}_B + \mathcal{L}_F + \mathcal{L}_{vgg} \tag{6}$$

4. Experiments

4.1. Implementation Details

We follow the similar idea in [34] to generate pairwised synthetic images with same style. We use over 1500 fonts and 10000 background images to generate a total of 1 million training images and 10000 test images. The input images are resized to 64 × 256 and the batch size is 32. All weights are initialized from a zero-mean normal distribution with a standard deviation of 0.01 The Adam optimizer [16] with $\beta_1 = 0.9$ and $\beta_2 = 0.999$ is used to optimize the whole framework. The learning rate is set to 0.0001 in the training phase. We implement our model under the TensorFlow framework [1]. Most modules of our method are GPU-accelerated.

4.2. Benchmark Datasets

We evaluate our proposed method on several public benchmark datasets.

IIIT 5K-Words [24] (IIIT5K) contains 3000 cropped word images for testing, while each image is assigned with a 50-word lexicon and a 1k-word lexicon. All images are collected from the Internet.

Street View Text [32] (SVT) is collected from Google Street View, which contains 647 images in the test set. Many images are severely corrupted by noise and blur, or have very low resolutions. Each image is associated with a 50-word lexicon.

ICDAR 2013 [14] (IC13) is obtained from the Robust Reading Chaallenges 2013. We follow the protocol proposed by [32], where images contain non-alphanumeric characters or those having less than three characters are not taken into consideration. After filtering samples, the dataset contains 857 images without any pre-defined lexicon.

ICDAR 2015 [13] (IC15) is more challenging than IC13, because most of the word images suffer from motion blur and low resolution. Moreover, many images contain severe geometric distortion, such as arbitrary oriented, perspective or curved texts. We filter images following the same protocol in IC13.

SVT-Perspective [27] (SVTP) contains 639 cropped images for testing, which are collected from side-view angle snapshots in Google Street View. Most of the images in SVT-Perspective are heavily deformed by perspective distortion.

CUTE80 [30] is collected for evaluating curved text recognition. It contains 288 cropped images for testing, which is selected from 80 high-resolution images taken in the natural scene.

4.3. Evaluation Metrics

We adopt the commonly used metrics in image generation to evaluate our method, which includes the following:

- MSE, also known as l2 error.
- PSNR, which computes the the ratio of peak signal to noise.
- SSIM, which computes the mean structural similarity index between two images
- Text Recognition Accuracy, we use text recognition model CRNN [31] to evaluate generated images.
- Text Detection Accuracy, we use text detection model EAST [43] to evaluate generated images.

A lower l2 error or higher SSIM and PSNR mean the results are similar to ground truth.

4.4. Ablation Studies

In this section, we empirically investigate how the performance of our proposed framework is affected by different model settings. Our study mainly focuses on these aspects: the content shape transformation network, the self-attention network, and dilated convolution in background completion network. Some qualitative results are presented in Figure 6

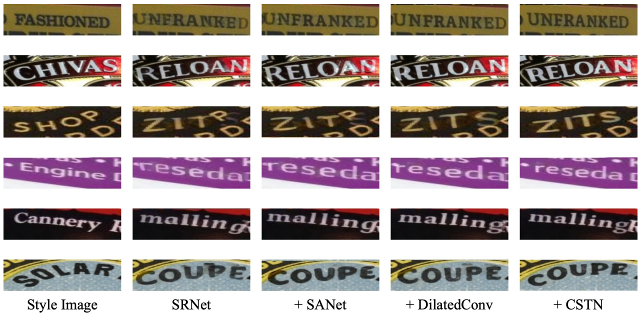


Figure 6. Some results of ablation study.

Content Shape Transformation Network (CSTN) Content shape transformation network (CSTN) aims to transform the content image according to the geometrical attributes of the style image. This is critical for text style transfer in real-world images, because scene text images often contain severe geometric distortion, such as in arbitrary oriented, perspective or curved form. With CSTN, the coherence of geometrical attributes between content and style images could be achieved. Although the whole model is difficult to train on real images, the CSTN can be finetuned on real datasets. As illustrated in Figure 6, the positions of generated text are more plausible. Quantitative results of CSTN is shown in Table 2, the PSNR increased by over 0.35 and SSIM increased by over 0.017 on average.

Self-attention Network Self-attention network is used to adequately combine the content features and style features. According to Table 2, with single level self-attention network, the average l_2 error is decreased by about 0.003, the average PSNR is increased by about 0.3, and the average SSIM is increased by about 0.012. To use more global statistics of the style and content features, we adopt a multi-level self-attention network to fuse global and local patterns. With multi-level self-attention network, all the metrics have been improved.

Dilated Convolution Dilated convolutional layers can enlarge the pixel regions to reconstruct the background images, therefore, it is easier to generate higher quality images. According to Table 2, the background completion network with dilated convolutional layers has a better performance on all metrics.

Table 2. Quantitative results on synthetic test dataset.

Method	English			Chinese		
	l_2	PSNR	SSIM	l_2	PSNR	SSIM
pix2pix [10]	0.0953	12.32	0.551	0.11531	10.09	0.3523
SRNet [34]	0.0472	14.91	0.6213	0.0512	14.77	0.5719
w/o CSTN	0.0436	15.22	0.6375	0.0463	14.98	0.5903
w/o SA	0.0421	15.31	0.6401	0.0459	15.02	0.5987
w/o DilatedConv	0.0402	15.23	0.6479	0.0432	15.15	0.6032
SwapText (single)	0.0397	15.53	0.6523	0.0422	15.38	0.6112
SwapText (multi)	**0.0381**	**16.04**	**0.6621**	**0.0420**	**15.46**	**0.6189**

4.5. Comparison with Prior Work

To evaluate our proposed method, we compared it with two types of text swapping method: pix2pix proposed in [10] and SRNet proposed by Wu *et al.* [34]. We use our generated datasets to train and test these two models. Both methods maintain the same configurations according to the papers.

Quantitative results In Table 2, we give some quantitative results of our method and other two competing methods. Clearly, our proposed method has a significant improvement on all the metrics across different languages. The average l_2 error is decreased by over 0.009, the average PSNR is increased by over 0.9, and the average SSIM is increased by over 0.04 than the second best method.

To further evaluate the quality of generated images, we propose to use text recognition and detection accuracy on generated images. We use the text recognition model CRNN to evaluate our generated images on SVT-P, IC13 and IC15 dataset. The CRNN model is trained on the mix of training images on these datasets and the recognition accuracy is present in Table 3. On IC13, the recognition accuracy is even higher than the real test set. We use an adapted version of EAST [43] to detect text in the images. Since the implementation of the original EAST is not available, we use the public implementation[1] with ResNet-50 backbone. We replace the texts in images of IC13 and IC15 test sets, then evaluate the generated datasets using the model trained on IC13 and IC15 training datasets. According to the comparison results presented in Table 4, the F-measure on generated IC13 and IC15 test sets are 78.4% and 80.2% respectively, which is close to the metrics on real test set. This indicates that the images generated by our framework are very realistic and can even fool the text detection model.

Table 3. Comparison of text recognition accuracy on real and generated images.

Dateset	SVT-P	IC13	IC15
Real	**54.3**	68.0	**55.2**
pix2pix	22.1	34.7	25.8
SRNet	48.7	66.8	50.2
Generated	54.1	**68.3**	54.9

Table 4. Comparison of text detection accuracy between real data and generated data on IC13 and IC15 datasets.

Test Set	IC13			IC15		
	R	P	F	R	P	F
Real	74.5	84.0	79.0	77.3	84.6	80.8
pix2pix	66.4	80.7	72.8	71.8	79.3	75.3
SRNet	70.4	82.9	76.1	74.2	82.5	78.1
SwapText	73.9	83.5	78.4	76.8	84.1	80.2

4.6. Image Based Text Translation

Image based translation is one of the most important applications of arbitrary text style transfer. In this section, we present some image based translation examples, which are illustrated in Figure 7. We conduct translation between english and chinese. According to the results, we can find that no matter the target language is chinese or english, the color, geometric deformation and background texture can be kept very well, and the structure of characters is the same as the input text.

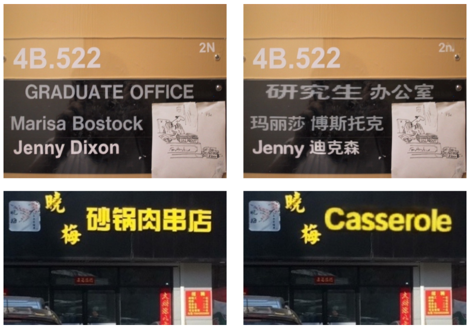


Figure 7. Image based translation examples. (*Left*) Input images. (*Right*) Translation results.

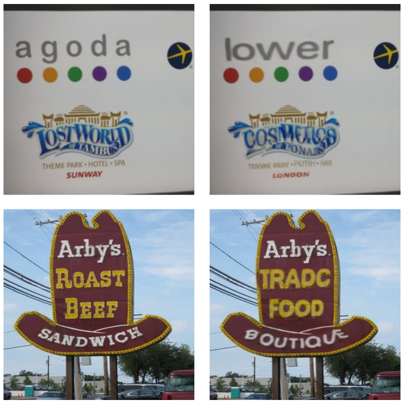


Figure 8. Failure Cases. (*Top*) Wavy text. (*Bottom*) WordArt.

In Figure 9, we also present some example results of our model evaluated on scene text datasets. According to Figure 9, our model can replace the text in the input image while keeping the original fonts, colors, sizes and background textures.

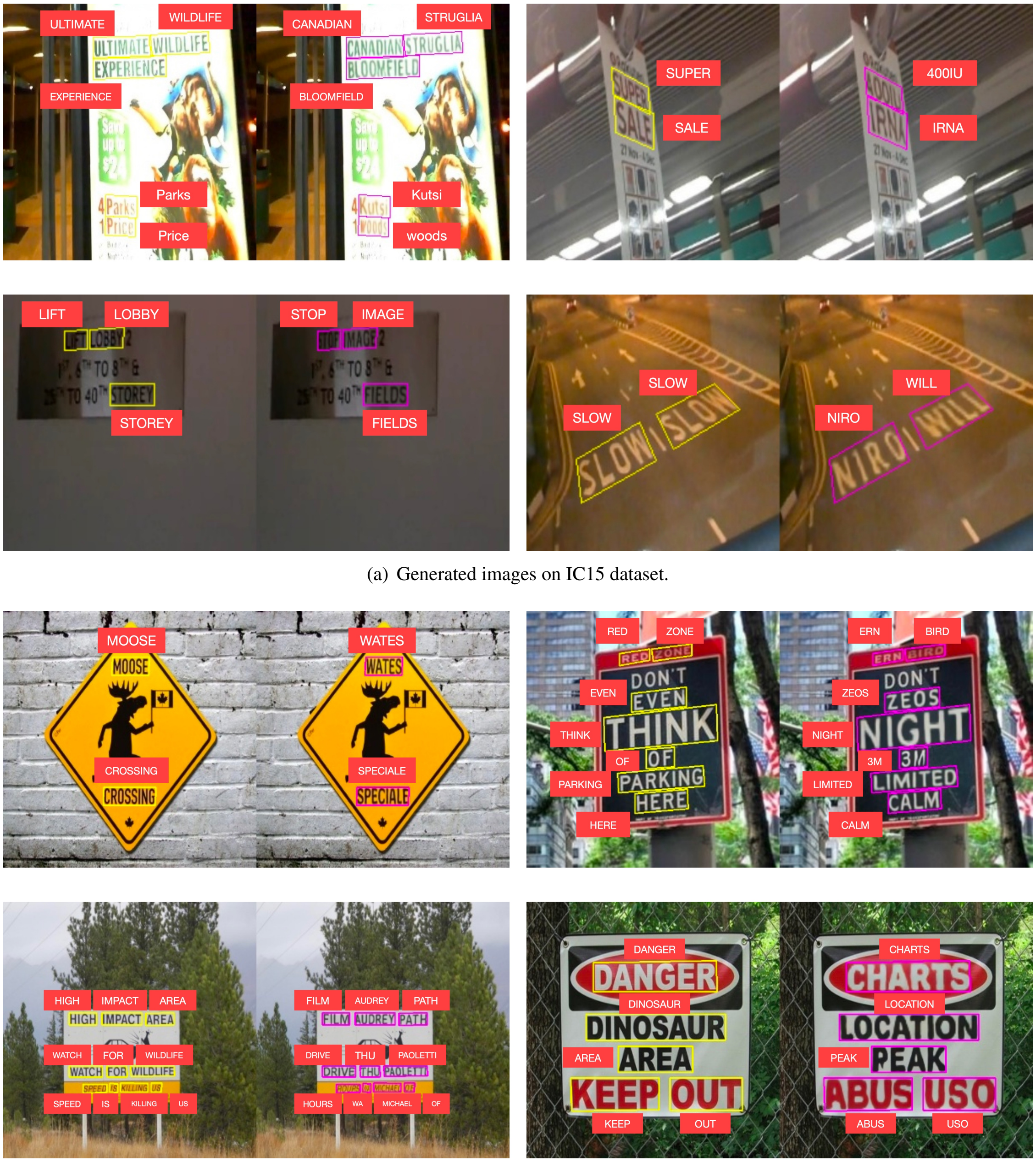
4.7. Limitations

Our method has the following limitations. Due to the limited amount of training data, the geometric attribute space and font space are not fully exploited. Our proposed method fails when the text in style image is waved, see Figure 8 (Top). Figure 8 (Bottom) shows a failure case on style image with WordArt.

5. Conclusion

In this study, we proposed a robust scene text swapping framework SwapText to address a novel task of replacing

[1] https://github.com/argman/EAST



(a) Generated images on IC15 dataset.

(b) Generated images on IC17 dataset.

Figure 9. Generated images on scene text datasets. The image on the left is the original image, while the right one is the generated image.

texts in the scene text images by intended texts. We adopt a divide-and-conquer strategy, decompose the problem into three sub-networks, namely text swapping network, background completion network and the fusion network. In the text swapping network, the features of content image and style image are extracted simultaneously and then combined by a self-attention network. To better learn the representation of content image, we use a Content Shape Transformation Network (CSTN) to transform the content image according to the geometrical attributes of the style image. Then, a background completion network is used to generate the background image of style image. Finally, the output of text swapping network and background completion network are fed into the fusion network to generate more realistic and semantically coherent images. Qualitative and quantitative results on several public scene text datasets demonstrate the superiority of our approach.

In the future work, we will explore to generate more controllable text images based on the fonts and colors.

References

[1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: A system for large-scale machine learning. In *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, pages 265–283, 2016.

[2] Samaneh Azadi, Matthew Fisher, Vladimir G Kim, Zhaowen Wang, Eli Shechtman, and Trevor Darrell. Multi-content gan for few-shot font style transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7564–7573, 2018.

[3] David Berthelot, Thomas Schumm, and Luke Metz. Began: Boundary equilibrium generative adversarial networks. *arXiv: Learning*, 2017.

[4] Paul Debevec. Rendering synthetic objects into real scenes: Bridging traditional and image-based graphics with global illumination and high dynamic range photography. In *ACM SIGGRAPH 2008 classes*, page 32. ACM, 2008.

[5] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. pages 2414–2423, 2016.

[6] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[7] Ankush Gupta, Andrea Vedaldi, and Andrew Zisserman. Synthetic data for text localisation in natural images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2315–2324, 2016.

[8] Xintong Han, Zuxuan Wu, Zhe Wu, Ruichi Yu, and Larry S Davis. Viton: An image-based virtual try-on network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7543–7552, 2018.

[9] Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa. Globally and locally consistent image completion. *ACM Transactions on Graphics (ToG)*, 36(4):107, 2017.

[10] Phillip Isola, Junyan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. *arXiv: Computer Vision and Pattern Recognition*, 2016.

[11] Max Jaderberg, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Synthetic data and artificial neural networks for natural scene text recognition. *arXiv preprint arXiv:1406.2227*, 2014.

[12] Levent Karacan, Zeynep Akata, Aykut Erdem, and Erkut Erdem. Learning to generate images of outdoor scenes from attributes and semantic layouts. *arXiv: Computer Vision and Pattern Recognition*, 2016.

[13] Dimosthenis Karatzas, Lluis Gomezbigorda, Anguelos Nicolaou, Suman K Ghosh, Andrew D Bagdanov, Masakazu Iwamura, Jiri Matas, Lukas Neumann, Vijay Ramaseshan Chandrasekhar, Shijian Lu, et al. Icdar 2015 competition on robust reading. pages 1156–1160, 2015.

[14] Dimosthenis Karatzas, Faisal Shafait, Seiichi Uchida, Masakazu Iwamura, Lluis Gomez I Bigorda, Sergi Robles Mestre, Joan Mas, David Fernandez Mota, Jon Almazan, and Lluis Pere De Las Heras. Icdar 2013 robust reading competition. pages 1484–1493, 2013.

[15] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *International Conference on Learning Representations*, 2018.

[16] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[17] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[18] Iryna Korshunova, Wenzhe Shi, Joni Dambre, and Lucas Theis. Fast face-swap using convolutional neural networks. *International Conference on Computer Vision*, pages 3697–3705, 2017.

[19] Christoph Lassner, Gerard Ponsmoll, and Peter V Gehler. A generative model of people in clothing. *International Conference on Computer Vision*, pages 853–862, 2017.

[20] Shangbang Long, Jiaqiang Ruan, Wenjie Zhang, Xin He, Wenhao Wu, and Cong Yao. Textsnake: A flexible representation for detecting text of arbitrary shapes. pages 19–35, 2018.

[21] Liqian Ma, Xu Jia, Qianru Sun, Bernt Schiele, Tinne Tuytelaars, and Luc Van Gool. Pose guided person image generation. *neural information processing systems*, pages 406–416, 2017.

[22] Liqian Ma, Qianru Sun, Stamatios Georgoulis, Luc Van Gool, Bernt Schiele, and Mario Fritz. Disentangled person image generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 99–108, 2018.

[23] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv: Learning*, 2014.

[24] Anand Mishra, Karteek Alahari, and CV Jawahar. Scene text recognition using higher order language priors. In *BMVC-British Machine Vision Conference*. BMVA, 2012.

[25] Aaron van den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. *arXiv preprint arXiv:1601.06759*, 2016.

[26] Dae Young Park and Kwang Hee Lee. Arbitrary Style Transfer with Style-Attentional Networks. *CoRR*, cs.CV, 2018.

[27] Trung Quy Phan, Palaiahnakote Shivakumara, Shangxuan Tian, and Chew Lim Tan. Recognizing text with perspective distortion in natural scenes. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 569–576, 2013.

[28] Amit Raj, Patsorn Sangkloy, Huiwen Chang, James Hays, Duygu Ceylan, and Jingwan Lu. Swapnet: Image based garment transfer. In *European Conference on Computer Vision*, pages 679–695. Springer, 2018.

[29] Scott E Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. Generative adversarial text to image synthesis. *International Conference on Machine Learning*, pages 1060–1069, 2016.

[30] Anhar Risnumawan, Palaiahankote Shivakumara, Chee Seng Chan, and Chew Lim Tan. A robust arbitrary text detection

system for natural scene images. *Expert Systems with Applications*, 41(18):8027–8048, 2014.

[31] Baoguang Shi, Xiang Bai, and Cong Yao. An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. *IEEE transactions on pattern analysis and machine intelligence*, 39(11):2298–2304, 2017.

[32] Kai Wang, Boris Babenko, and Serge Belongie. End-to-end scene text recognition. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 1457–1464. IEEE, 2011.

[33] Yi Wang, Xin Tao, Xiaojuan Qi, Xiaoyong Shen, and Jiaya Jia. Image inpainting via generative multi-column convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 329–338, 2018.

[34] Liang Wu, Chengquan Zhang, Jiaming Liu, Junyu Han, Jingtuo Liu, Errui Ding, and Xiang Bai. Editing Text in the Wild. *arXiv.org*, Aug. 2019.

[35] Mingkun Yang, Yushuo Guan, Minghui Liao, Xin He, Kaigui Bian, Song Bai, Cong Yao, and Xiang Bai. Symmetry-constrained Rectification Network for Scene Text Recognition. *CoRR*, cs.CV, 2019.

[36] Shuai Yang, Zhangyang Wang, Zhaowen Wang, Ning Xu, Jiaying Liu, and Zongming Guo. Controllable Artistic Text Style Transfer via Shape-Matching GAN. *arXiv.org*, May 2019.

[37] Xi Yin, Xiang Yu, Kihyuk Sohn, Xiaoming Liu, and Manmohan Chandraker. Towards large-pose face frontalization in the wild. *International Conference on Computer Vision*, pages 4010–4019, 2017.

[38] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Generative image inpainting with contextual attention. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5505–5514, 2018.

[39] Fangneng Zhan, Shijian Lu, and Chuhui Xue. Verisimilar image synthesis for accurate detection and recognition of texts in scenes. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 249–266, 2018.

[40] Fangneng Zhan, Chuhui Xue, and Shijian Lu. Ga-dan: Geometry-aware domain adaptation network for scene text detection and recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9105–9115, 2019.

[41] Fangneng Zhan, Hongyuan Zhu, and Shijian Lu. Spatial fusion gan for image synthesis. pages 3653–3662, 2018.

[42] Han Zhang, Ian Goodfellow, Dimitris N Metaxas, and Augustus Odena. Self-attention generative adversarial networks. *arXiv: Machine Learning*, 2018.

[43] Xinyu Zhou, Cong Yao, He Wen, Yuzhi Wang, Shuchang Zhou, Weiran He, and Jiajun Liang. East: An efficient and accurate scene text detector. *Computer Vision and Pattern Recognition*, pages 2642–2651, 2017.